Sri Sathya Sai Institute of Higher Learning

(Deemed to be University)

MDSC-206-PROJECT

Little known ways to data analysis with Spotify data

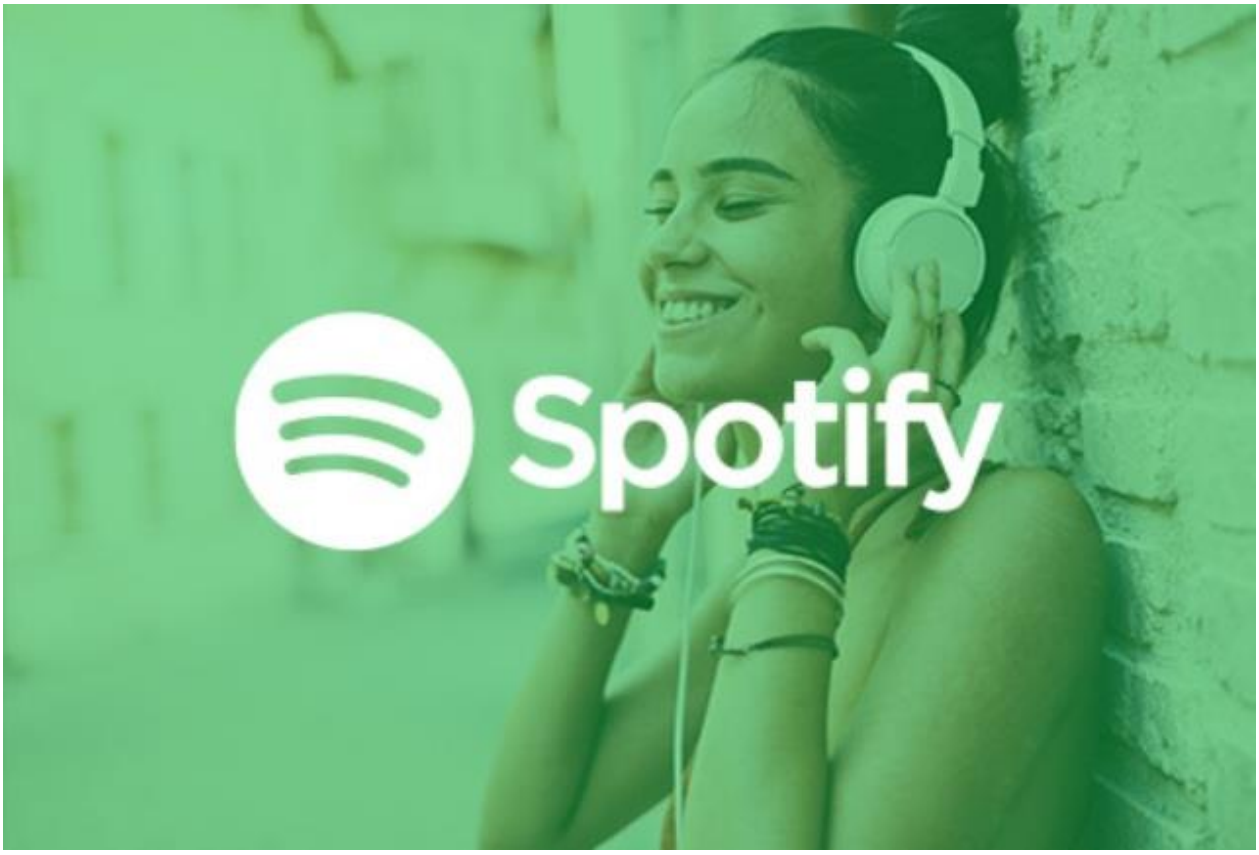
Krishnakanth G

20233

Contents

Introduction

I was assigned the task of exploring any dataset that piqued my interest. I went for the Spotify dataset. Spotify is the most popular music streaming service on the planet. Users of the service need only register to gain access to one of the world's largest music libraries, as well as podcasts and other audio material.



While exploring it in innumerable ways viz Exploratory Data Analysis, Linear models, Logistic models, etc., the inferences I accumulated were mind-boggling.

Now let's jump in and look at what I explored through this dataset and what are my inferences.

Data Description

The data consists of 7000 observations and 19 attributes. All the attributes are defined below:

variable	Description
Acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
Artists	The person who composed the song.
Danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability,

	beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
Duration_ms	Duration of song in milliseconds.
Energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
explicit	Explicit represent any offensive content in the song. 1 represent offensive content whereas 0 represent non-offensive content.
Id	Song ID which is unique.
Instrumentalness	Predicts whether a track contains no vocals. 'Ooh' and 'aah' sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly 'vocal'. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
Key	All keys on octave encoded as values ranging from 0 to 11, starting on C as 0, C# as 1 and so on.
liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
loudness	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.
Mode	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
Name	Name of the song.
Popularity	Song popularity ranges from 0 to 100 where higher is better.
Release_date	Date of song released.

Speechiness	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g., talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
Tempo	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
Valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g., happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g., sad, depressed, angry).
Year	Year of the release of song.

First six observations from data are shown below:

```
##        acousticness                              artists danceability duration_ms
## 119130     0.000143                          ['Slade']        0.582      326507
## 133294     0.357000 ['New Riders of the Purple Sage']        0.557      312533
## 42053      0.943000                    ['Lead Belly']        0.572       48640
## 90493      0.205000                 ['Jack Johnson']        0.850      158413
## 94329      0.992000              ['Ignacio Corsini']        0.698      168400
## 100972     0.797000                  ['Marvin Gaye']        0.443      239862
##        energy explicit                      id instrumentalness key liveness
## 119130  0.833        0 0NXjn6R9lGplKOzZUNpOui         2.08e-03   2   0.0472
## 133294  0.300        0 1wEHFEIfJA8hdPbGvLYZG9         0.00e+00   7   0.2800
## 42053   0.199        0 1rm0dPDX0kGjAmxiQ4vzCf         0.00e+00   9   0.1930
## 90493   0.834        0 422qOiUJZCd7SaEyoPAEnQ         1.74e-01   1   0.0837
## 94329   0.192        0 6k6oWqs8cZnnjY38LdP08T         5.77e-03   2   0.2430
## 100972  0.330        0 2g42UTWeZmlRuzrOJnao0O         2.28e-05   1   0.1260
##        loudness mode                        name popularity release_date
## 119130   -7.266    1  Run Runaway - 12" Version         30   1983-12-03
## 133294  -15.061    1          Last Lonely Eagle         21         1971
## 42053   -14.951    1  Bring Me Li'l Water Silvy          1   1939-09-09
## 90493    -5.771    0               Rodeo Clowns         54   2003-01-01
## 94329   -17.343    1 La Carreta - Remasterizado          0   1928-05-14
## 100972  -12.964    1    If I Should Die Tonight         34   1973-08-28
##        speechiness   tempo valence year
## 119130      0.0501 129.118   0.725 1983
## 133294      0.0333 114.223   0.362 1971
## 42053       0.0695 200.486   0.963 1939
## 90493       0.1050  90.974   0.771 2003
## 94329       0.1280  78.218   0.681 1928
## 100972      0.0322 137.905   0.359 1973
```

Data cleaning

Checking for Duplicate rows

```
##  [1] acousticness     artists          danceability     duration_ms
##  [5] energy           explicit         id               instrumentalness
##  [9] key              liveness         loudness         mode
## [13] name             popularity       release_date     speechiness
## [17] tempo            valence          year
## <0 rows> (or 0-length row.names)
```

The output shows that there are no duplicates in the dataset.

Checking for Missing values

```
## [1] 0
```

Here we are checking for any missing values in the dataset. But from above output it is clear that there are no missing values.

Adding new variable using existing variables

Here we are adding new variable namely duration_min which is produced from duration_ms. we are converting milli seconds into minutes and storing it in new variable.

Removing unnecessary columns

```
##   acousticness       artists           danceability     duration_ms
##  Min.   :0.00000   Length:7000        Min.   :0.0000   Min.   :   5108
##  1st Qu.:0.07727   Class :character   1st Qu.:0.4150   1st Qu.: 166067
##  Median :0.50400   Mode  :character   Median :0.5465   Median : 206320
##  Mean   :0.49256                      Mean   :0.5344   Mean   : 234787
##  3rd Qu.:0.89000                      3rd Qu.:0.6630   3rd Qu.: 265300
##  Max.   :0.99600                      Max.   :0.9690   Max.   :4254375
##      energy          explicit            id            instrumentalness
##  Min.   :0.0000   Min.   :0.00000   Length:7000        Min.   :0.000000
##  1st Qu.:0.2480   1st Qu.:0.00000   Class :character   1st Qu.:0.000000
##  Median :0.4750   Median :0.00000   Mode  :character   Median :0.000534
##  Mean   :0.4861   Mean   :0.06486                      Mean   :0.195001
##  3rd Qu.:0.7173   3rd Qu.:0.00000                      3rd Qu.:0.243750
##  Max.   :1.0000   Max.   :1.00000                      Max.   :0.990000
##       key            liveness         loudness           mode
##  Min.   : 0.000   Min.   :0.0000   Min.   :-60.000   Min.   :0.000
##  1st Qu.: 2.000   1st Qu.:0.0991   1st Qu.:-14.838   1st Qu.:0.000
##  Median : 5.000   Median :0.1380   Median :-10.790   Median :1.000
##  Mean   : 5.215   Mean   :0.2103   Mean   :-11.729   Mean   :0.697
##  3rd Qu.: 8.000   3rd Qu.:0.2740   3rd Qu.: -7.410   3rd Qu.:1.000
##  Max.   :11.000   Max.   :0.9970   Max.   : -0.117   Max.   :1.000
##      name             popularity    release_date        speechiness
##  Length:7000        Min.   : 0.00   Length:7000        Min.   :0.00000
```

```
##  Class :character   1st Qu.: 2.00   Class :character   1st Qu.:0.03500
##  Mode  :character   Median :26.00   Mode  :character   Median :0.04500
##                     Mean   :26.12                      Mean   :0.10522
##                     3rd Qu.:43.00                      3rd Qu.:0.07575
##                     Max.   :93.00                      Max.   :0.96800
##      tempo           valence            year       duration_min
##  Min.   :  0.00   Min.   :0.0000   Min.   :1920   Min.   : 0.08513
##  1st Qu.: 94.77   1st Qu.:0.3060   1st Qu.:1956   1st Qu.: 2.76778
##  Median :116.52   Median :0.5390   Median :1978   Median : 3.43867
##  Mean   :117.70   Mean   :0.5264   Mean   :1978   Mean   : 3.91312
##  3rd Qu.:135.26   3rd Qu.:0.7452   3rd Qu.:1999   3rd Qu.: 4.42167
##  Max.   :221.11   Max.   :1.0000   Max.   :2021   Max.   :70.90625
```

Removing the unnecessary columns like id, duration_ms and release date.

Datatype Conversion

we can observe the variables mode, explicit and key are integers but actually they are factors. So, convert them to factors.

```
## Rows: 7,000
## Columns: 17
## $ acousticness     <dbl> 1.43e-04, 3.57e-01, 9.43e-01, 2.05e-01, 9.92e-01, ...
## $ artists          <chr> "['Slade']", "['New Riders of the Purple Sage']", ...
## $ danceability     <dbl> 0.582, 0.557, 0.572, 0.850, 0.698, 0.443, 0.847, 0...
## $ energy           <dbl> 0.8330, 0.3000, 0.1990, 0.8340, 0.1920, 0.3300, 0....
## $ explicit         <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,...
## $ instrumentalness <dbl> 2.08e-03, 0.00e+00, 0.00e+00, 1.74e-01, 5.77e-03, ...
## $ key              <fct> 2, 7, 9, 1, 2, 1, 11, 10, 5, 9, 11, 1, 3, 0, 2, 5,...
## $ liveness         <dbl> 0.0472, 0.2800, 0.1930, 0.0837, 0.2430, 0.1260, 0....
## $ loudness         <dbl> -7.266, -15.061, -14.951, -5.771, -17.343, -12.964...
## $ mode             <fct> 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,...
## $ name             <chr> "Run Runaway - 12\" Version", "Last Lonely Eagle",...
## $ popularity       <int> 30, 21, 1, 54, 0, 34, 47, 18, 54, 53, 0, 57, 7, 20...
## $ speechiness      <dbl> 0.0501, 0.0333, 0.0695, 0.1050, 0.1280, 0.0322, 0....
## $ tempo            <dbl> 129.118, 114.223, 200.486, 90.974, 78.218, 137.905...
## $ valence          <dbl> 0.725, 0.362, 0.963, 0.771, 0.681, 0.359, 0.845, 0...
## $ year             <int> 1983, 1971, 1939, 2003, 1928, 1973, 2002, 1961, 19...
## $ duration_min     <dbl> 5.4417833, 5.2088833, 0.8106667, 2.6402167, 2.8066...
```

After converting the variables. we can observe datatypes are correct.

Data splitting

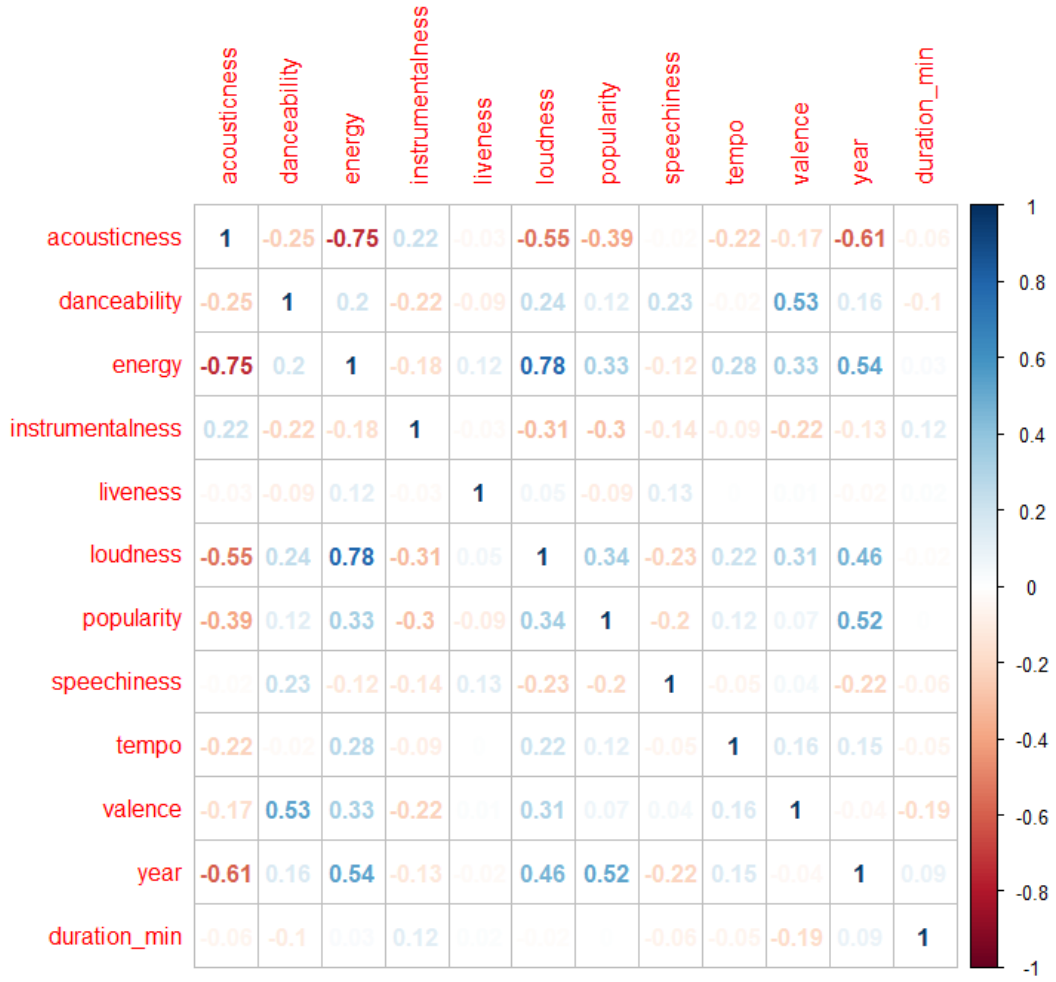
```
## [1] 5000   17

## [1] 2000   17
```

Here we are splitting data as 5000 observations for train set and 200 observations for test set randomly.

Exploratory Data Analysis

Correlation plot



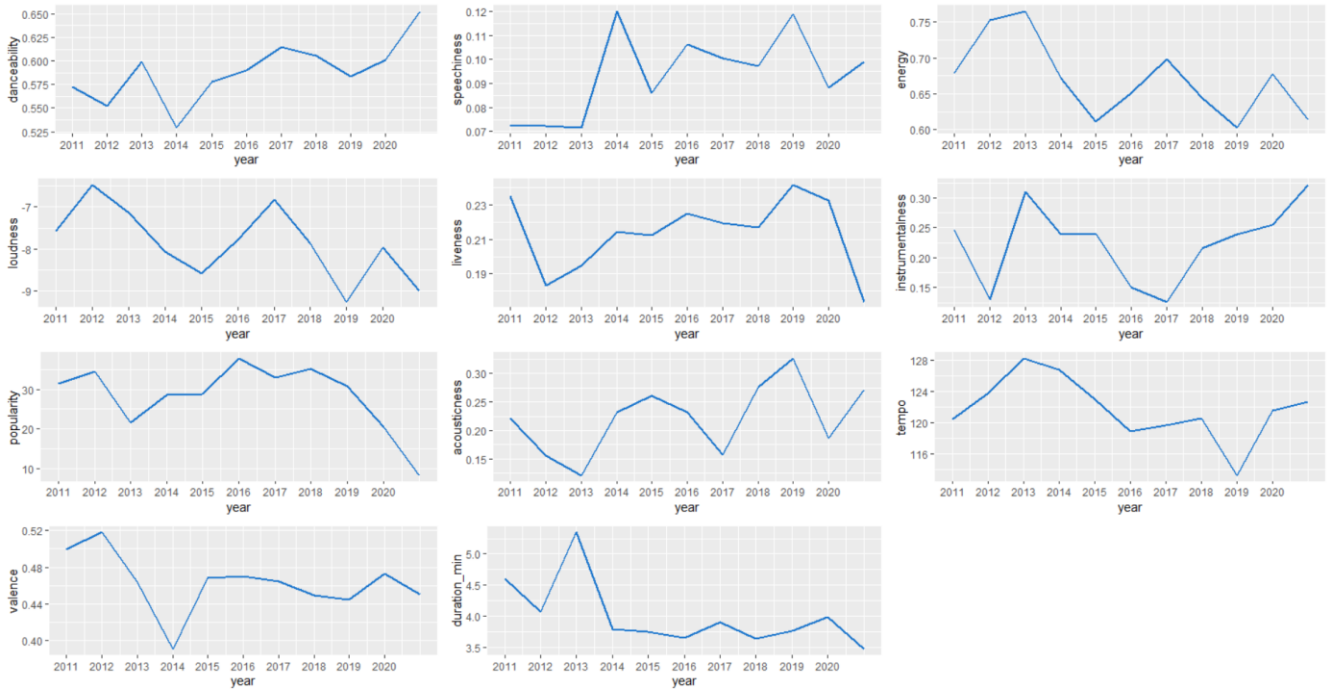
	acousticness	danceability	energy	instrumentalness	liveness	loudness	popularity	speechiness	tempo	valence	year	duration_min
acousticness	1	-0.25	-0.75	0.22	-0.03	-0.55	-0.39	-0.02	-0.22	-0.17	-0.61	-0.06
danceability	-0.25	1	0.2	-0.22	-0.09	0.24	0.12	0.23	-0.02	0.53	0.16	-0.1
energy	-0.75	0.2	1	-0.18	0.12	0.78	0.33	-0.12	0.28	0.33	0.54	0.03
instrumentalness	0.22	-0.22	-0.18	1	0.03	-0.31	-0.3	-0.14	-0.09	-0.22	-0.13	0.12
liveness	-0.03	-0.09	0.12	-0.03	1	0.05	-0.09	0.13	0.01	0.01	-0.02	0.02
loudness	-0.55	0.24	0.78	-0.31	0.05	1	0.34	-0.23	0.22	0.31	0.46	0.02
popularity	-0.39	0.12	0.33	-0.3	-0.09	0.34	1	-0.2	0.12	0.07	0.52	0
speechiness	-0.02	0.23	-0.12	-0.14	0.13	-0.23	-0.2	1	-0.05	0.04	-0.22	-0.06
tempo	-0.22	-0.02	0.28	-0.09	0.01	0.22	0.12	-0.05	1	0.16	0.15	-0.05
valence	-0.17	0.53	0.33	-0.22	0.01	0.31	0.07	0.04	0.16	1	-0.04	-0.19
year	-0.61	0.16	0.54	-0.13	0.02	0.46	0.52	-0.22	0.15	-0.04	1	0.09
duration_min	-0.06	-0.1	0.03	0.12	0.02	-0.02	0	-0.06	-0.05	-0.19	0.09	1

From the above correlation plot, we can infer the following:

- ✓ acousticness and energy are highly negatively correlated.
- ✓ loudness and energy are highly positively correlated.
- ✓ acousticness and popularity are negatively correlated.
- ✓ acousticness and year are negatively correlated.
- ✓ popularity and year are positively correlated.
- ✓ loudness and year are positively correlated.
- ✓ acousticness and loudness are negatively correlated.
- ✓ danceabiltity and valence are positively correlated.
- ✓ energy and year are positively correlated.

Other variables are somehow related but not very good.

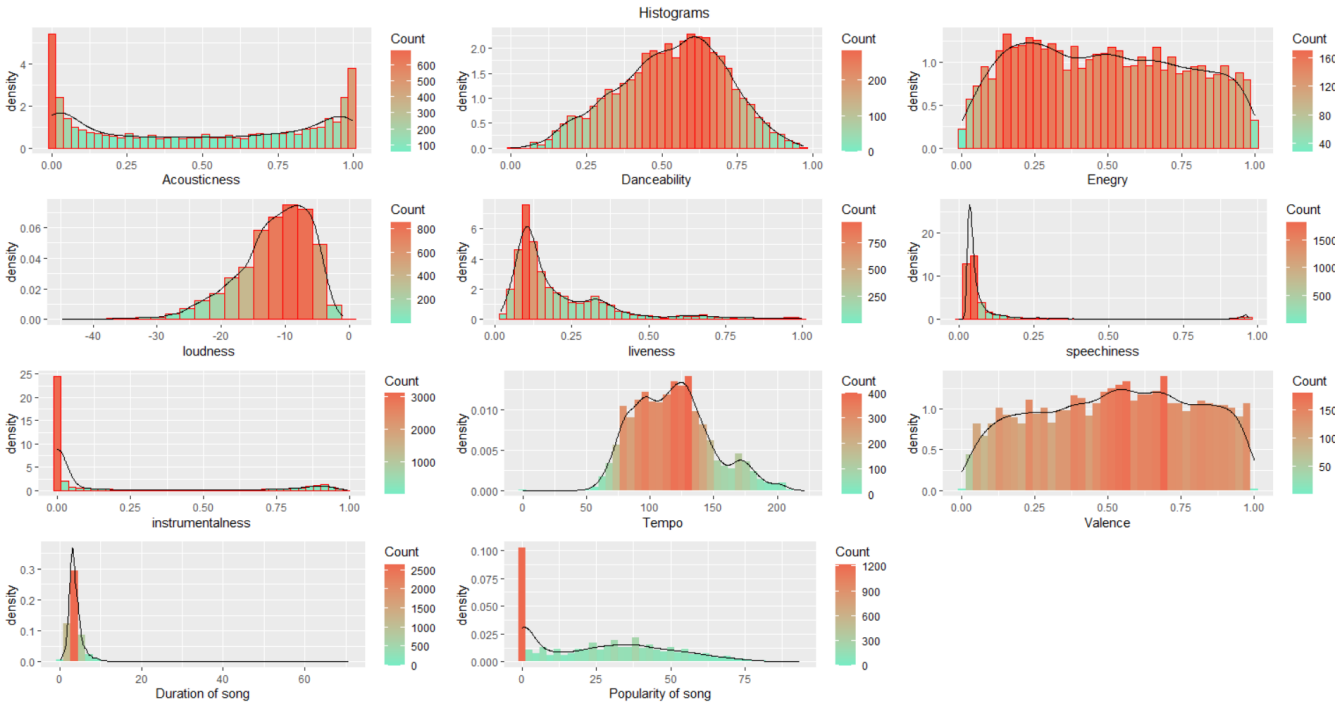
Trends over last decade



From the above Trends plot we can infer the following from year 2011 to 2020:

- ✓ Trends in danceability are increased across the years except in 2014 and 2019.
- ✓ Trends in speechiness are fluxing over the years except in 2011-2013.
- ✓ Trends in energy are coming down over the years compared to 2013.
- ✓ Trends in loudness are coming down over the years compared to 2012.
- ✓ Trends in instrumentalness are increased after 2012 and drastically decreased at 2017. After 2017 it is increasing over the years.
- ✓ Trends in popularity are coming down over the years compared to 2016.
- ✓ Trends in acousticness are fluxing over the years in an increasing manner.
- ✓ Trends in tempo come down till 2019 and went up in 2020.
- ✓ Trends in valence come down till 2014 and then increased slowly over the years except in 2019.
- ✓ Trends in duration of song decreased over the years compared to initial years.

Histograms



From the histograms of the continuous variables, we can infer the following:

- ✓ The mean of Acousticness is around 0.5. Its density curve looks like not normal.
- ✓ The mean of Danceability is around 0.54. It is skewed towards its left and density curve looks like normal.
- ✓ The mean of Energy is around 0.48. Its density curve looks like normal.
- ✓ The mean of loudness is around -11. It is skewed towards its left and density curve looks like not normal.
- ✓ The mean of liveness is around 0.20. It is skewed towards its right and density curve looks like not normal.
- ✓ The mean of speechiness is around 0.1. It is highly skewed towards its right and density curve looks like not normal.
- ✓ The mean of instrumentalness is around 0.1. It is highly skewed towards its right and density curve looks like not normal.
- ✓ The mean of Tempo is around 117. Its density curve looks like normal.
- ✓ The mean of Valence is around 0.5. Its density curve looks like normal.
- ✓ The mean of Popularity of song is around 26. It is highly skewed towards its right.
- ✓ The mean of Duration of song is around 3.9. It is highly skewed towards its right.

The output shows that the density curves of all continuous variables. we can observe that all the curve are somewhat skewed from which it is clear that continuous variables may not be normal,

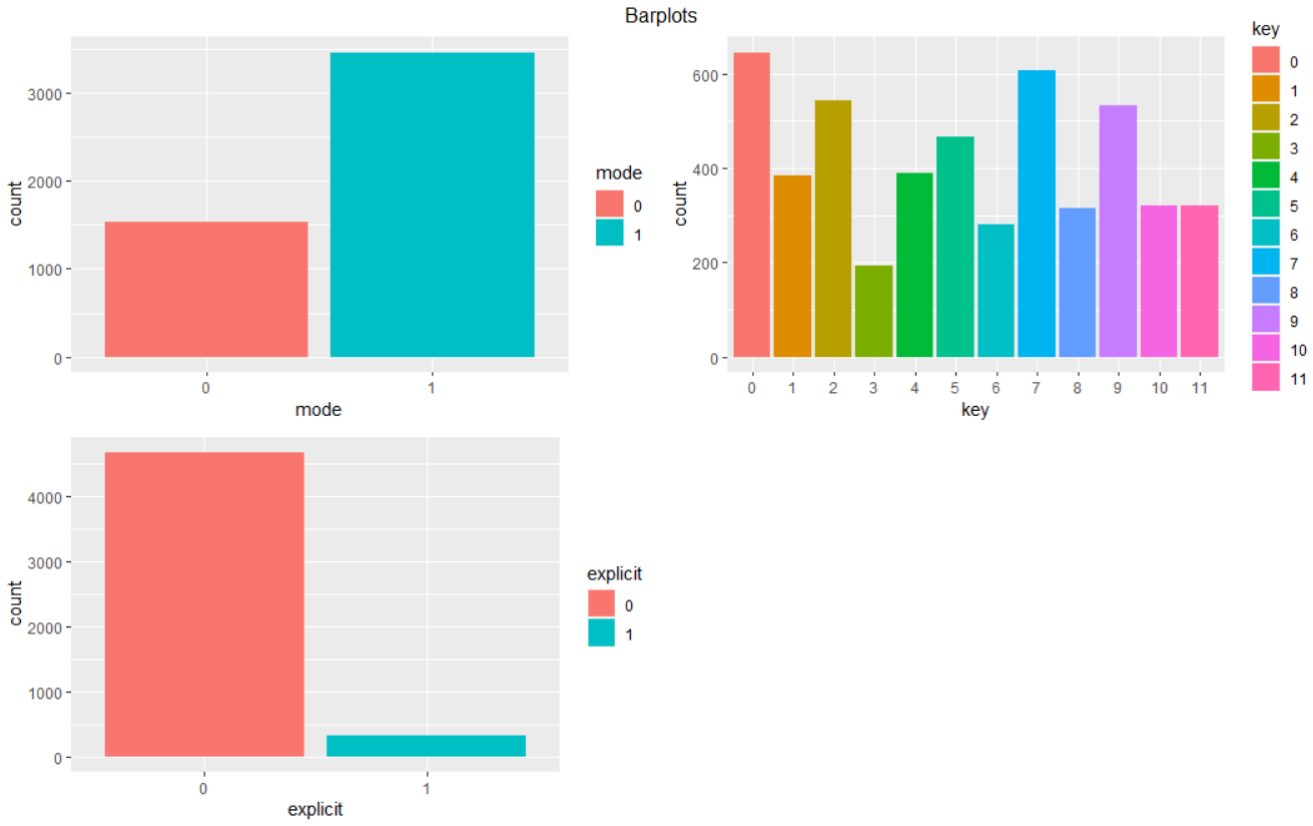
but we can't tell that for sure we have to check statistically and for that we have shapiro test which will statically signify the normality.

Shapiro test

```
## [1] "No variable is Normal in the data with 5% significance level"
```

The output shows that there are no normal variables in the data at 5% significance level. Now we have strong evidence for saying that no variable is normal in the data.

Bar plots

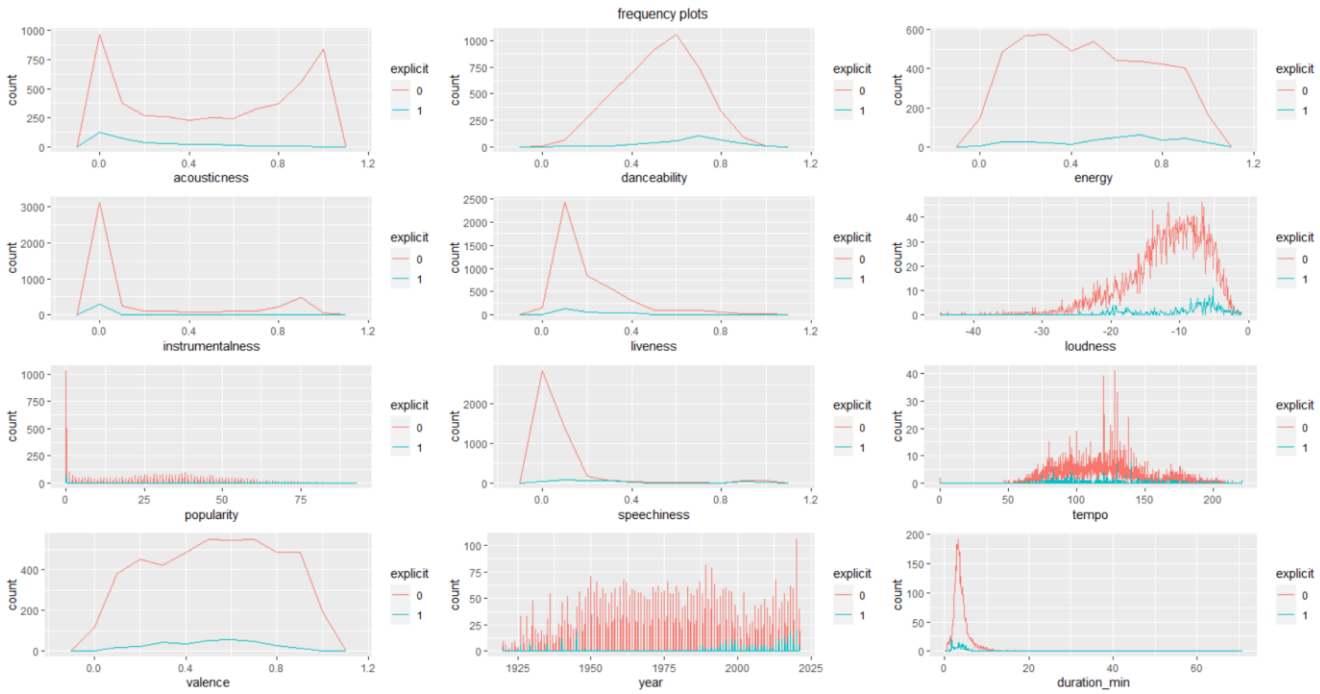


From the above Bar plots, we can infer the following:

- ✓ The frequency of modes having major (1) and minor (0) of a track are 3466 and 1534.
- ✓ The frequency of keys having 0,1,2,3,4,5,6,7,8,9,10 and 11 are 645, 385, 543, 193, 390, 467, 281, 607, 316, 533, 320, and 320.
- ✓ The frequency of explicit having 0 and 1 are 4674 and 326.

Most of the songs in the data set are non-explicit means they are not discriminating anyone and many songs have modality as major which means the scale of melody is higher. Key of a song is the group of pitches and most of the songs have C (0) as key.

Frequency plots

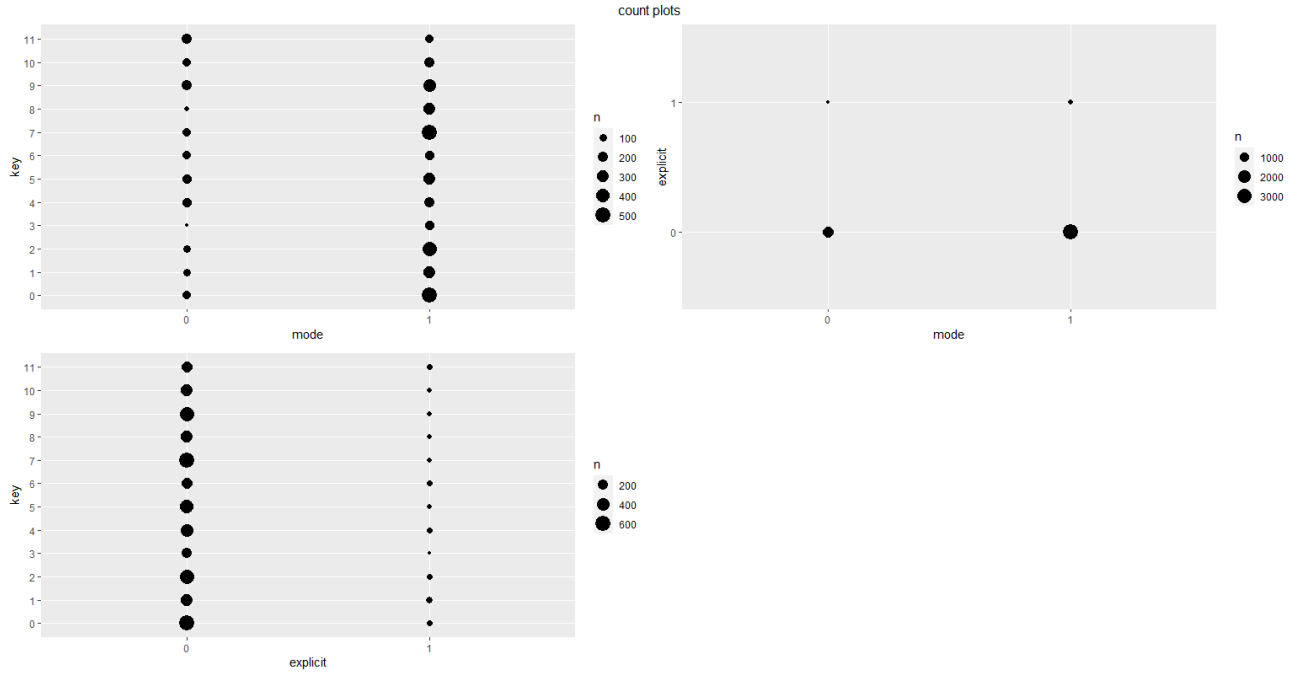


From the frequency plots, we can infer the following:

- ✓ The frequency of 0 and 1 in acousticness are high for non-explicit content.
- ✓ The frequency of 0.6 in danceabiltity is high for non-explicit content.
- ✓ The frequency of 0.1 to 0.9 in energy are high for non-explicit content.
- ✓ The frequency of 0 and 0.9 in instrumentalness is high and moderate for non-explicit content.
- ✓ The frequency of 0.1 and 0.2 in liveness is high and moderate for non-explicit content.
- ✓ The frequency of -20 to 0 in loudness are slowly going from low to high and them high to low for non-explicit content.
- ✓ The frequency of 0 in popularity is very high for non-explicit content.
- ✓ The frequency of 0 in speechiness is very high for non-explicit content.
- ✓ The frequency of tempo of songs is fluxing from 50 to 200 but it is very high at 128 and 122 for non-explicit content.
- ✓ The frequency of 0.1 to 0.9 in valence are high for non-explicit content.
- ✓ The frequency of number of songs in year is fluxing but it is very high at 2020 for non-explicit content.
- ✓ The frequency of duration of song is high for short length songs compared to long length

The Explicit content is very less compared to non-explicit content and it is very rare for all frequency plots.

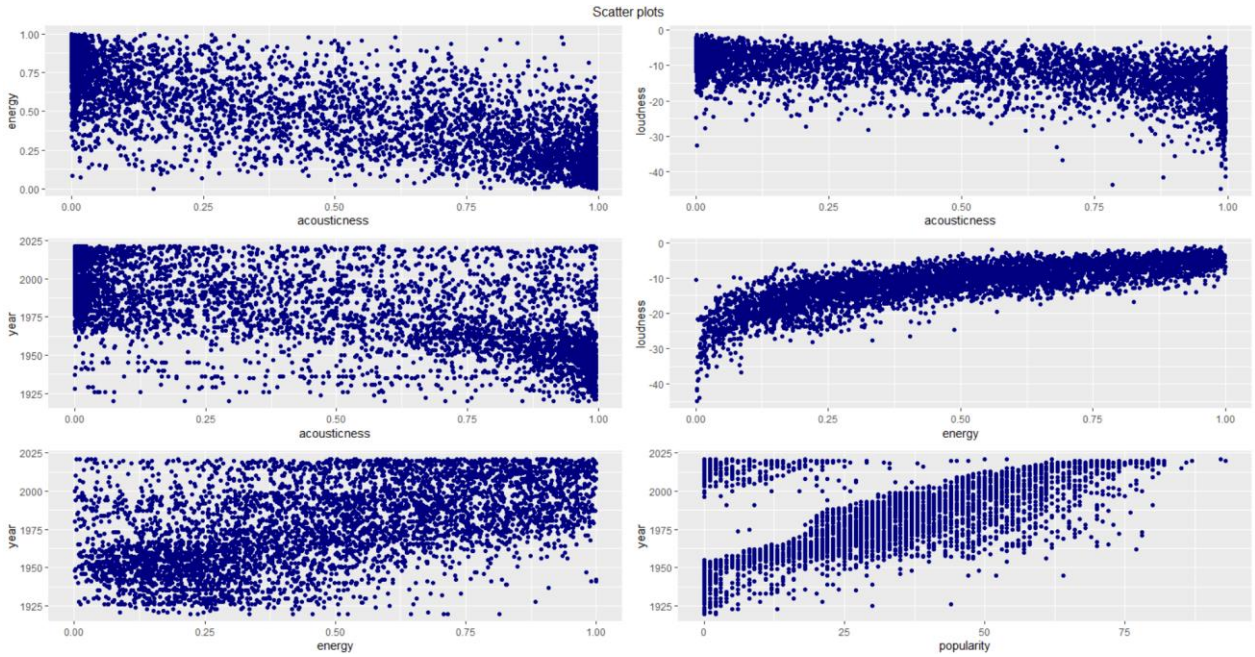
Count plots



From the above Count plots, we can infer the following:

- ✓ The count of major mode (1) and keys is more compared to minor mode (0) and keys.
- ✓ The count of mode and non-explicit content (0) is more compared to mode and explicit content (1).
- ✓ The count of non-explicit content (0) and keys is more compared to explicit content (1) and keys.

Scatter plots



From the above Scatter plots, we can infer the following:

- ✓ The acousticness vs energy plot shows that acousticness and energy are highly negatively correlated.
- ✓ The acousticness vs loudness plot shows that acousticness and loudness are negatively correlated.
- ✓ The acousticness vs year plot shows that acousticness and year are negatively correlated.
- ✓ The energy vs loudness plot shows that energy and loudness are highly positively correlated.
- ✓ The energy vs year plot shows that energy and year are positively correlated.
- ✓ The popularity vs year plot shows that popularity and year are positively correlated.

Cross tables

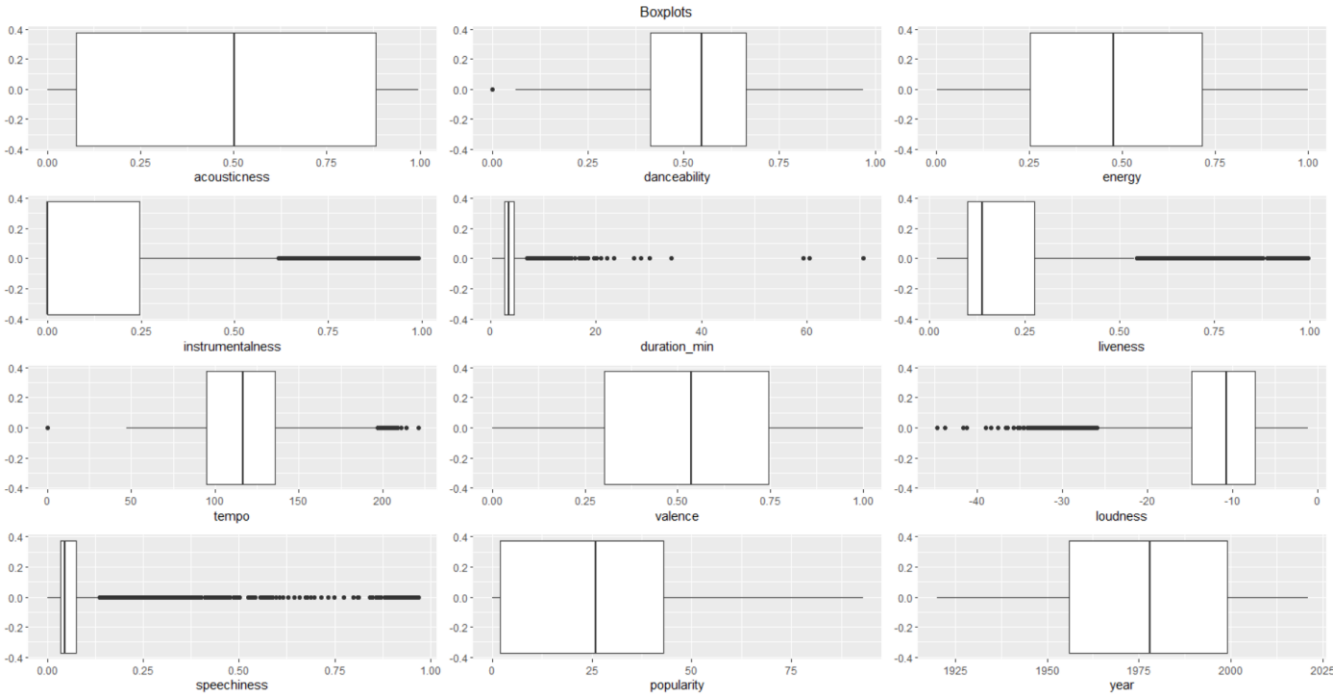
```
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
## Total Observations in Table:  5000
##
##              | train$explicit
##   train$mode |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1396 |       138 |      1534 |
##              |     1.006 |    14.425 |           |
##              |     0.910 |     0.090 |     0.307 |
##              |     0.299 |     0.423 |           |
##              |     0.279 |     0.028 |           |
## -------------|-----------|-----------|-----------|
##            1 |      3278 |       188 |      3466 |
##              |     0.445 |     6.384 |           |
##              |     0.946 |     0.054 |     0.693 |
##              |     0.701 |     0.577 |           |
##              |     0.656 |     0.038 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      4674 |       326 |      5000 |
##              |     0.935 |     0.065 |           |
## -------------|-----------|-----------|-----------|
```

The output shows the cross table between mode and explicit variables, we can infer the following:

- ✓ The songs which are minor of track and have non-explicit content are 1396 and mean within mode and explicit variables is 0.279.

- ✓ The songs which are minor of track and have explicit content are 138 and mean within mode and explicit variables is 0.028.
- ✓ The songs which are major of track and have non-explicit content are 3278 and mean within mode and explicit variables is 0.656.
- ✓ The songs which are major of track and have explicit content are 188 and mean within mode and explicit variables is 0.038.

Boxplots



From the above box plots, we can infer the following:

- ✓ The acousticness plot is almost symmetric with no outliers.
- ✓ The danceability plot is almost symmetric with some outliers.
- ✓ The energy plot is almost symmetric with no outliers.
- ✓ The duration_min plot is highly skewed to its right with many outliers.
- ✓ The liveness plot is highly skewed to its right with many outliers.
- ✓ The tempo plot is skewed to its right with many outliers.
- ✓ The valence plot is almost symmetric with no outliers.
- ✓ The loudness plot is highly skewed to its left with many outliers.
- ✓ The speechiness plot is highly skewed to its right with many outliers.
- ✓ The popularity plot is highly skewed to its right with no outliers.
- ✓ The year plot is highly skewed to its left with no outliers.

Principal component analysis

In simple words, PCA is a method of obtaining important variables in form of components from a large set of variables available in a data set. It extracts low dimensional set of features by taking a projection of relevant dimensions from a high dimensional data set with a motive to capture as much information as possible.

Correlation matrix

```
##                  acousticness danceability      energy instrumentalness
## acousticness       1.00000000  -0.24887619 -0.74953751       0.22484808
## danceability      -0.24887619   1.00000000  0.19914113      -0.21947960
## energy            -0.74953751   0.19914113  1.00000000      -0.17905609
## instrumentalness   0.22484808  -0.21947960 -0.17905609       1.00000000
## liveness          -0.03146358  -0.08551877  0.11819936      -0.02568150
## loudness          -0.54768996   0.24109663  0.78460871      -0.31483005
## popularity        -0.39409234   0.12374238  0.33457773      -0.29878257
## speechiness       -0.01736748   0.23385540 -0.12449509      -0.13527629
## tempo             -0.22355917  -0.02215603  0.27519495      -0.08534684
## valence           -0.17464001   0.53442554  0.33435123      -0.22184988
## year              -0.60797392   0.15955543  0.53727408      -0.12689866
## duration_min      -0.05747502  -0.10155932  0.02552291       0.12008932
##                     liveness    loudness    popularity speechiness
## acousticness     -0.031463584 -0.54768996 -0.3940923402 -0.01736748
## danceability     -0.085518765  0.24109663  0.1237423841  0.23385540
## energy            0.118199364  0.78460871  0.3345777332 -0.12449509
## instrumentalness -0.025681499 -0.31483005 -0.2987825692 -0.13527629
## liveness          1.000000000  0.05472879 -0.0904186601  0.12879171
## loudness          0.054728786  1.00000000  0.3440021886 -0.22554279
## popularity       -0.090418660  0.34400219  1.0000000000 -0.20413434
## speechiness       0.128791708 -0.22554279 -0.2041343410  1.00000000
## tempo             0.001058046  0.21993994  0.1183078713 -0.05103275
## valence           0.011955804  0.30805248  0.0690012711  0.04339294
## year             -0.024324797  0.46487192  0.5244149374 -0.21751897
## duration_min      0.017280805 -0.01792090  0.0009950746 -0.06055318
##                         tempo     valence        year  duration_min
## acousticness     -0.223559166 -0.17464001 -0.60797392 -0.0574750184
## danceability     -0.022156030  0.53442554  0.15955543 -0.1015593201
## energy            0.275194947  0.33435123  0.53727408  0.0255229076
## instrumentalness -0.085346837 -0.22184988 -0.12689866  0.1200893226
## liveness          0.001058046  0.01195580 -0.02432480  0.0172808046
## loudness          0.219939936  0.30805248  0.46487192 -0.0179209050
## popularity        0.118307871  0.06900127  0.52441494  0.0009950746
## speechiness      -0.051032747  0.04339294 -0.21751897 -0.0605531848
## tempo             1.000000000  0.15726394  0.15390824 -0.0516763142
## valence           0.157263940  1.00000000 -0.03643434 -0.1877530723
## year              0.153908244 -0.03643434  1.00000000  0.0876265152
## duration_min     -0.051676314 -0.18775307  0.08762652  1.0000000000
```

The above is the correlation matrix from which we can say something about the relationship between all the variables. The Eigen values and Eigen vectors for this matrix are determined below:

Eigen values and vectors

```
##  [1] 3.5831125 1.7174198 1.1659687 1.0743931 0.9843951 0.8807778 0.7848809
##  [8] 0.6415000 0.4332550 0.3440951 0.2682939 0.1219081

##              [,1]        [,2]        [,3]        [,4]        [,5]         [,6]
##  [1,]  0.42996348  0.06934499 -0.16849462 -0.15991347  0.05254584  0.163432461
##  [2,] -0.20951509  0.49586505 -0.16764042  0.25693405 -0.33578257  0.002708364
##  [3,] -0.45823819 -0.05538009  0.22092417 -0.12563261 -0.15885044  0.058201825
##  [4,]  0.21556126 -0.26864244  0.09750492 -0.20621479 -0.58580103 -0.012978018
##  [5,] -0.01296211  0.04844728  0.75601259 -0.07205259  0.23097060  0.411184531
##  [6,] -0.43429973 -0.02482359  0.06213537 -0.15241542 -0.07115299  0.209441769
##  [7,] -0.31417253 -0.18139866 -0.31594244  0.22722588  0.33261608  0.038068709
##  [8,]  0.08237593  0.43761036  0.35381265  0.40460512  0.14940792 -0.437797580
##  [9,] -0.18012278 -0.02455614  0.10051628 -0.56922048  0.13926888 -0.698961546
## [10,] -0.21560546  0.50342798 -0.09297174 -0.26144376 -0.31067730  0.137093246
## [11,] -0.37609579 -0.29588228 -0.05605540  0.22119756  0.03139305 -0.027818699
## [12,]  0.01155509 -0.32421088  0.25076529  0.40626640 -0.45587785 -0.235704079
##              [,7]        [,8]        [,9]        [,10]       [,11]        [,12]
##  [1,]  0.26432559 -0.16115456  0.14814614  0.497381750  0.46964256  0.375699152
##  [2,] -0.06502149 -0.30545052  0.45263920  0.206557519 -0.36803606  0.151059543
##  [3,] -0.09566887  0.25916805 -0.21594404  0.049901718  0.01683887  0.753666161
##  [4,] -0.52329836 -0.29199789 -0.23435131  0.238345578 -0.00875440 -0.122848812
##  [5,]  0.05946147 -0.41800162  0.10675880 -0.013638889 -0.08457324 -0.036345412
##  [6,]  0.12188909  0.36103560  0.08467400  0.599507792  0.07353110 -0.466431037
##  [7,]  0.07171948 -0.50471978 -0.50624699  0.270042873 -0.12454320  0.015821623
##  [8,] -0.25723189  0.10556048 -0.25725726  0.261255429  0.28990183 -0.055165284
##  [9,]  0.14433731 -0.25247976  0.17207079  0.079330290 -0.06641179 -0.001663746
## [10,]  0.23170992 -0.20284653 -0.34477385 -0.335296332  0.38966202 -0.172620097
## [11,] -0.28662808 -0.21974729  0.42522380 -0.176460111  0.61148920 -0.029302632
## [12,]  0.62970432 -0.06449497 -0.03593087  0.004281389  0.03668147 -0.009752179
```

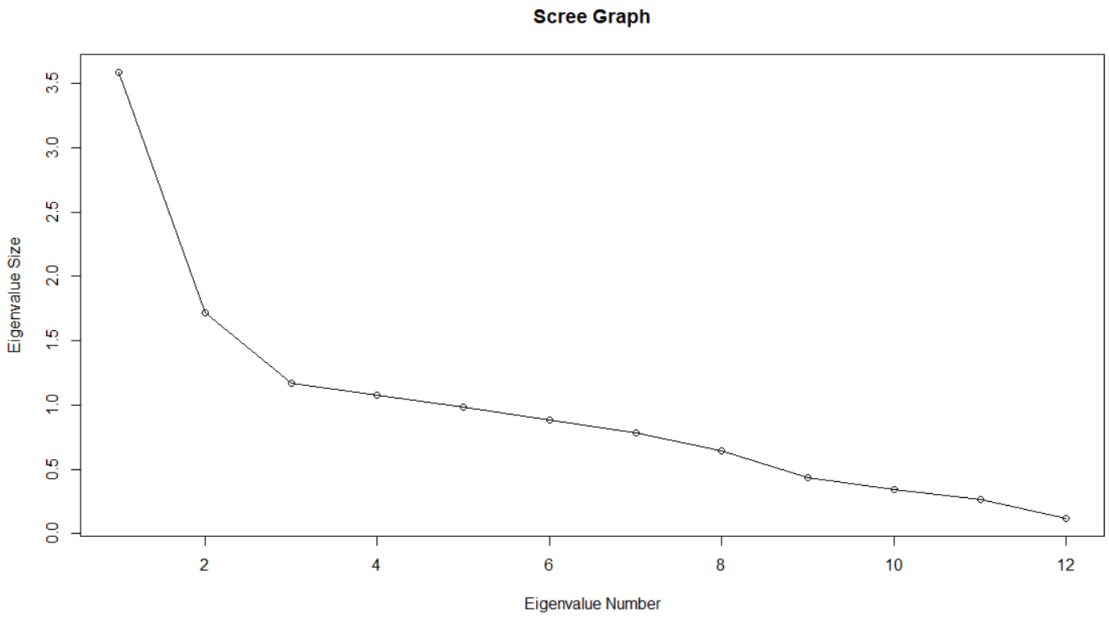
Principal components

We aim to find the components which explain the maximum variance. This is because, we want to retain as much information as possible using these components. So, higher is the explained variance, higher will be the information contained in those components.

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5    PC6     PC7
## Standard deviation     1.8929 1.3105 1.07980 1.03653 0.99217 0.9385 0.88594
## Proportion of Variance 0.2986 0.1431 0.09716 0.08953 0.08203 0.0734 0.06541
## Cumulative Proportion  0.2986 0.4417 0.53888 0.62841 0.71044 0.7838 0.84925
##                           PC8    PC9    PC10    PC11    PC12
## Standard deviation     0.80094 0.6582 0.58660 0.51797 0.34915
## Proportion of Variance 0.05346 0.0361 0.02867 0.02236 0.01016
## Cumulative Proportion  0.90270 0.9388 0.96748 0.98984 1.00000
```

This shows that the first principal component describes 29.8% of the variance, the second component describes 14.31% of the variance, the third component describes 9.7% of the variance, and so on. So, how do we figure out how many components to use in the modeling stage? A scree plot is a solution. To access the components that explain the most variability in the data, a scree plot is used.

scree plot

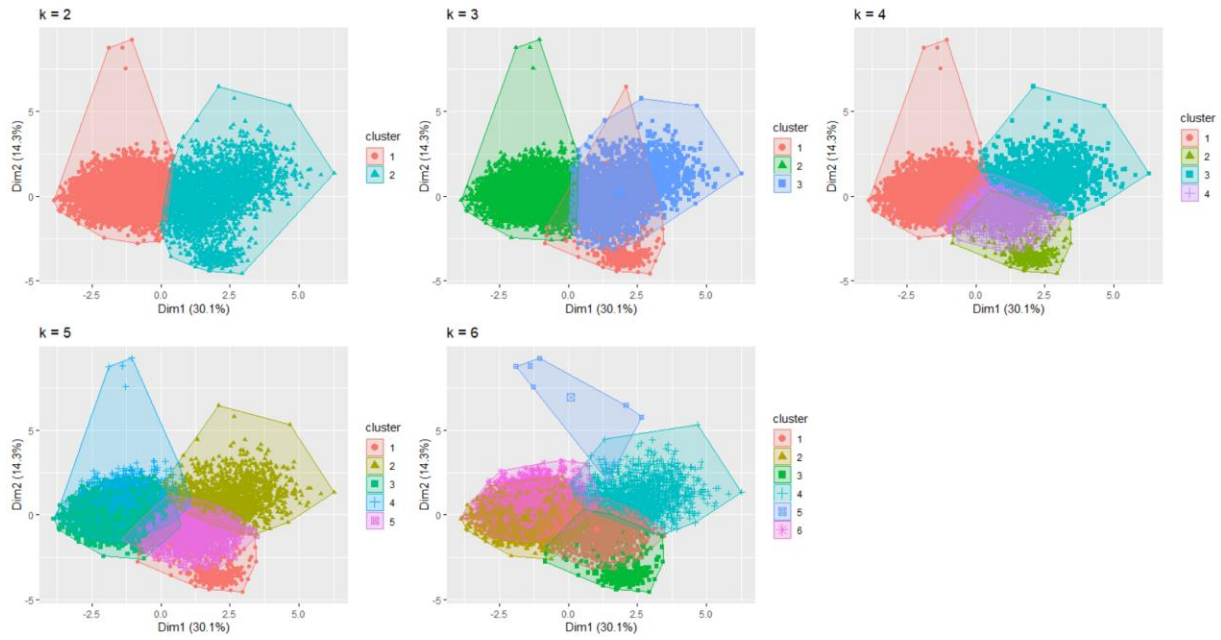


The graph above reveals that eight components contribute about 90% of the variability in the data set. In other words, we were able to reduce 12 predictors to 8 predictors using PCA without compromising explained variance. This is the PCA's strength.

K-means clustering

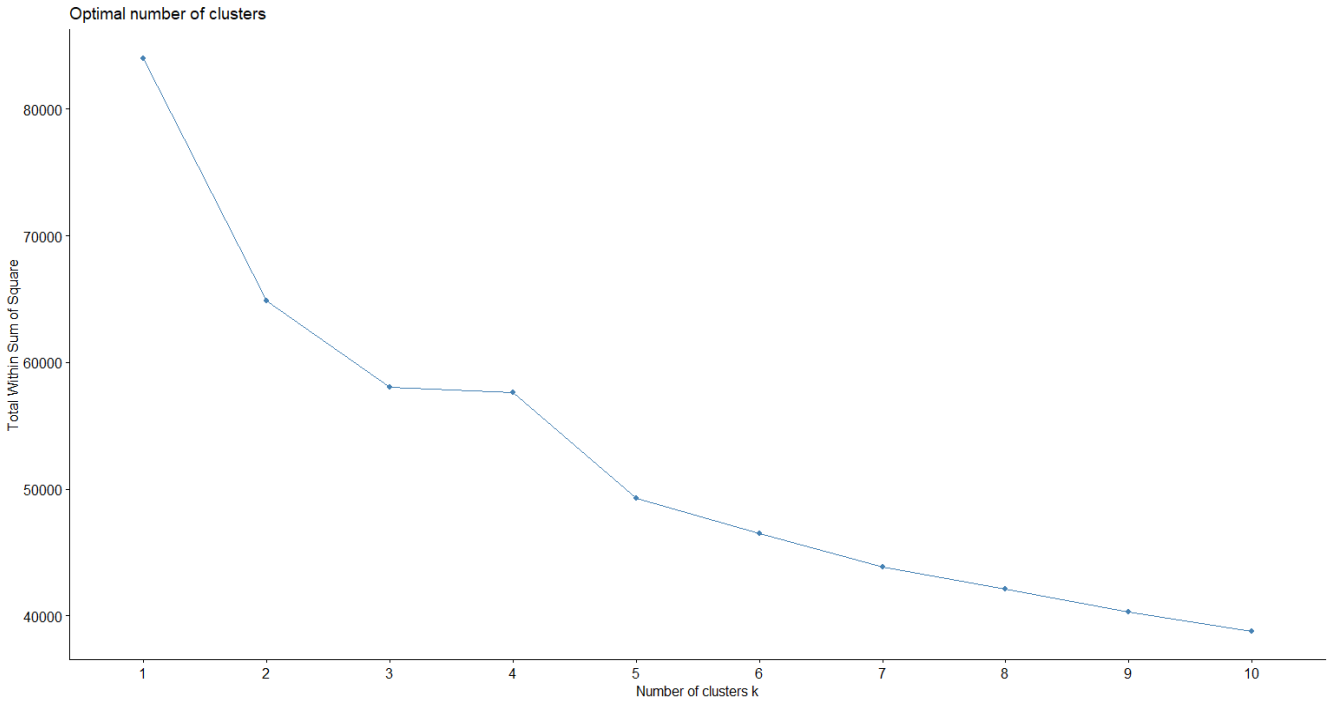
K-means clustering is a centroid-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, we associate each cluster with a centroid. The principal aim of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid. Now let's fit multiple k-means models by increasing the number of clusters.

Clustering



Above, we fitted the k-means with different k values and we also have the plots of all those clusters. Now how to find the best out of all those clusters. The best way to do that is the Elbow method, which is widely used to determine an optimal number of clusters in K-means clustering. This method takes into consideration the total within-cluster sum of square (WSS) as a function of the number of clusters K. Optimal value of K is such that adding another cluster does not improve much better the total WSS.

Determining and Visualizing the Optimal Number of Clusters



Looking at the above elbow curve, we can say that the optimal value of k is 4. We can dwell deep into clustering but instead of applying a unsupervised algorithms let's try with supervised algorithms and observe the results.

Linear models

choosing energy as dependent variable, we will find the best linear model to test the following hypothesis:

H_0: All β_i are equal to zero for all i

H_1: At least one $\beta_i \neq \beta_j$ where i≠j

Significance level: .05

Now let's try to find the best linear model with good adjusted R-squared and less AIC & BIC values.

Linear model1

Model : energy ~.

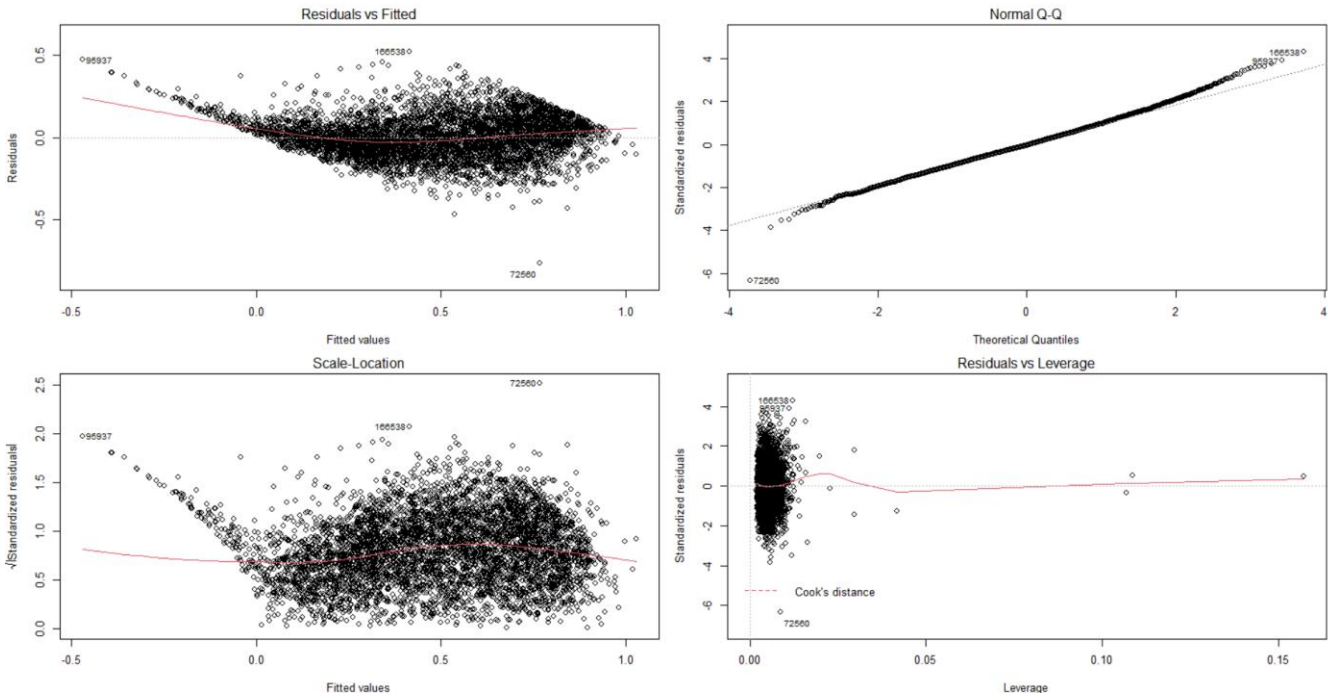
```
##
## Call:
## lm(formula = energy ~ ., data = train[, -c(2, 11)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76684 -0.07850 -0.00437  0.07568  0.52034
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -9.038e-01  1.916e-01  -4.716 2.47e-06 ***
## acousticness     -3.142e-01  6.599e-03 -47.619  < 2e-16 ***
## danceability     -1.798e-01  1.291e-02 -13.929  < 2e-16 ***
## explicit1        -4.188e-02  7.862e-03  -5.327 1.04e-07 ***
## instrumentalness  8.963e-02  5.929e-03  15.117  < 2e-16 ***
## key1              1.775e-02  7.908e-03   2.245   0.0248 *
## key2              1.576e-02  7.100e-03   2.219   0.0265 *
## key3              1.014e-02  1.002e-02   1.012   0.3117
## key4              9.407e-03  7.869e-03   1.195   0.2320
## key5              8.506e-03  7.425e-03   1.146   0.2520
## key6              6.427e-03  8.794e-03   0.731   0.4649
## key7              6.715e-03  6.886e-03   0.975   0.3295
## key8              1.585e-02  8.367e-03   1.894   0.0583 .
## key9              1.020e-02  7.165e-03   1.424   0.1545
## key10             1.295e-02  8.355e-03   1.549   0.1214
## key11             1.788e-02  8.474e-03   2.110   0.0349 *
## liveness          1.007e-01  1.017e-02   9.896  < 2e-16 ***
## loudness          2.452e-02  4.120e-04  59.525  < 2e-16 ***
## mode1            -5.528e-03  3.886e-03  -1.423   0.1549
## popularity       -1.308e-04  9.861e-05  -1.327   0.1847
## speechiness       6.082e-02  1.208e-02   5.034 4.98e-07 ***
## tempo             2.985e-04  6.008e-05   4.968 7.00e-07 ***
## valence           1.883e-01  8.571e-03  21.964  < 2e-16 ***
## year              8.816e-04  9.620e-05   9.165  < 2e-16 ***
## duration_min      1.364e-03  7.074e-04   1.928   0.0539 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1217 on 4975 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.8021
## F-statistic: 845.4 on 24 and 4975 DF,  p-value: < 2.2e-16
```

From the summary, we can conclude that the linear model is

$$enegry = -9.038 * e^{-1} - 3.142 * e^{-1} * acousticness - 1.798 * e^{-1} * danceability + \cdots \ldots + \epsilon$$

The adj R-squared value is 0.80 which is very good but we need to check the linearity of residuals vs fitted line and normality of the residuals.

Plots



Inference from the plots:

- **Residuals vs Fitted:** This plot is showing the residuals are having some nonlinear patterns.
- **Normal Q-Q plot:** This plot shows that residuals are not normally distributed.
- **Scale-Location plot:** This plot shows that Variance is almost constant.
- **Residual vs Leverage:** This plot talks about influential point.

Shapiro-Wilk normality test

```
##
##  Shapiro-Wilk normality test
##
## data:  linear_model1$residuals
## W = 0.99622, p-value = 5.251e-10

## [1] 0.0002384
```

From the output of the Shapiro test we can statistically prove something about normality. The p-value of Shapiro test is less than 0.05. so, we say that the residuals are not normal. And also, the model contains many non-significant variables. let's remove them and create a new model.

Linear model2

Model: energy ~ acousticness + danceability + instrumentalness + liveness + loudness + speechiness + tempo + valence + year + duration_min

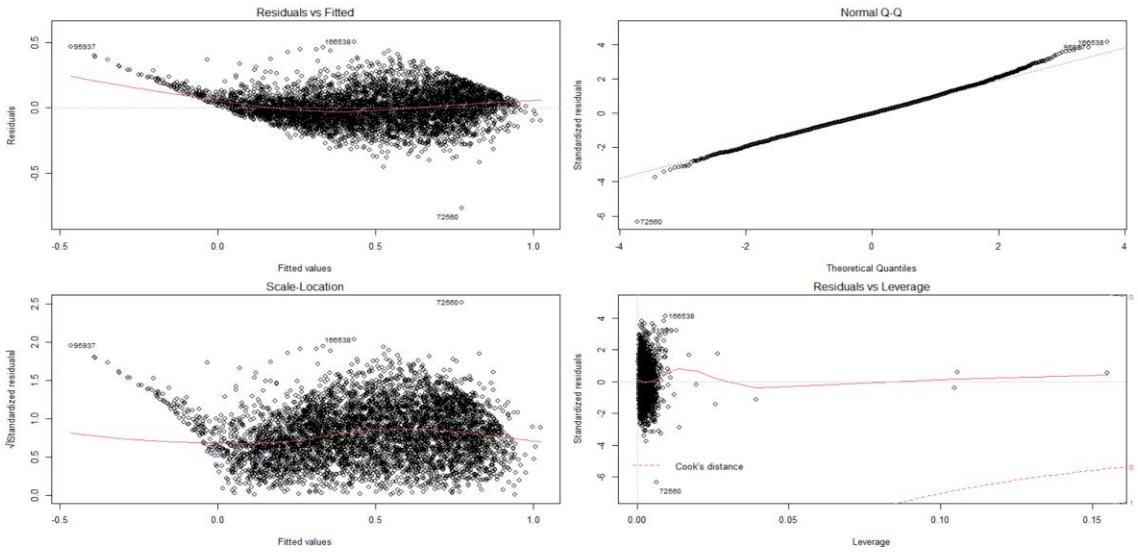
```
##
## Call:
## lm(formula = energy ~ acousticness + danceability + instrumentalness +
##     liveness + loudness + speechiness + tempo + valence + year +
##     duration_min, data = train[, -c(2, 11)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77073 -0.07920 -0.00411  0.07720  0.50234
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -7.307e-01  1.812e-01  -4.033 5.58e-05 ***
## acousticness     -3.125e-01  6.536e-03 -47.818  < 2e-16 ***
## danceability     -1.857e-01  1.285e-02 -14.449  < 2e-16 ***
## instrumentalness  9.257e-02  5.706e-03  16.221  < 2e-16 ***
## liveness          1.008e-01  1.013e-02   9.953  < 2e-16 ***
## loudness          2.436e-02  4.102e-04  59.377  < 2e-16 ***
## speechiness       4.551e-02  1.113e-02   4.088 4.42e-05 ***
## tempo             2.995e-04  6.018e-05   4.976 6.70e-07 ***
## valence           1.929e-01  8.504e-03  22.680  < 2e-16 ***
## year              7.930e-04  9.030e-05   8.782  < 2e-16 ***
## duration_min      1.651e-03  7.074e-04   2.334   0.0196 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.122 on 4989 degrees of freedom
## Multiple R-squared:  0.8014, Adjusted R-squared:  0.801
## F-statistic:  2013 on 10 and 4989 DF,  p-value: < 2.2e-16
```

From the summary, we can conclude that the linear model is

$$energy = -7.307 * e^{-1} - 3.125 * e^{-1} * acousticness - 1.857 * e^{-1} * danceability + \cdots \ldots + \epsilon$$

The adj R-squared value is still 0.80 which is very good but we need to check the linearity of residuals vs fitted line and normality of the residuals.

Plots



Inference from the plots:

- **Residuals vs Fitted:** This plot is showing the residuals are having some nonlinear patterns.
- **Normal Q-Q plot:** This plot shows that residuals are not normally distributed.
- **Scale-Location plot:** This plot shows that Variance is almost constant.
- **Residual vs Leverage:** This plot talks about influential point.

Shapiro-Wilk normality test

```
##
##  Shapiro-Wilk normality test
##
## data:  linear_model2$residuals
## W = 0.99638, p-value = 1.051e-09

## [1] 0.0001297
```

From the output of the Shapiro test we can see that the p-value is less than 0.05. so, we say that the residuals are not normal. This model does not contain any non-significant variables but normality and linearity are failed. So, lets normalize the model using square root transformation and then see the model.

Normalized linear model2

```
##
## Call:
## lm(formula = energyn ~ acousticness + danceability + instrumentalness +
##     liveness + loudness + speechiness + tempo + valence + year +
##     duration_min, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84424 -0.05548  0.00031  0.05609  0.39592
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.743e-01  1.330e-01   1.311 0.190006
## acousticness     -2.253e-01  4.798e-03 -46.954  < 2e-16 ***
## danceability     -1.078e-01  9.436e-03 -11.419  < 2e-16 ***
## instrumentalness  5.740e-02  4.189e-03  13.703  < 2e-16 ***
## liveness          7.625e-02  7.438e-03  10.252  < 2e-16 ***
## loudness          2.152e-02  3.011e-04  71.477  < 2e-16 ***
## speechiness       3.609e-02  8.173e-03   4.416 1.03e-05 ***
## tempo             2.377e-04  4.418e-05   5.381 7.74e-08 ***
## valence           1.630e-01  6.243e-03  26.108  < 2e-16 ***
## year              3.825e-04  6.629e-05   5.771 8.36e-09 ***
## duration_min      1.948e-03  5.192e-04   3.751 0.000178 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

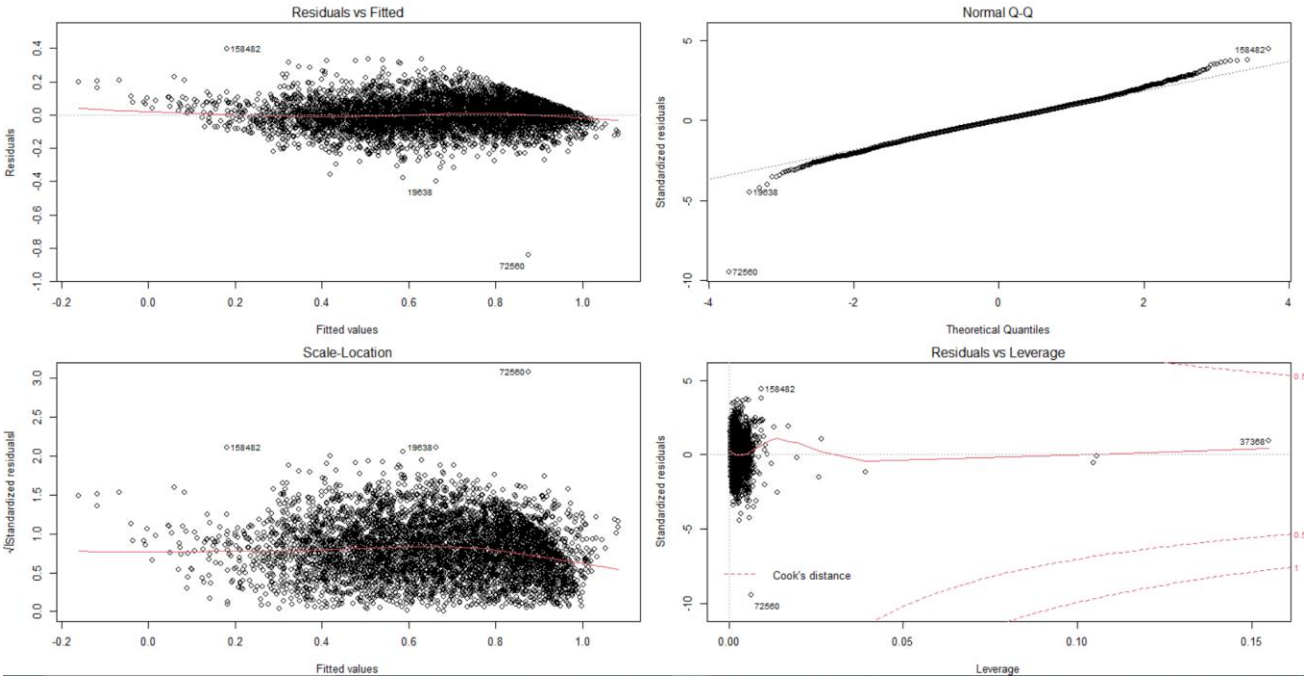
```
## Residual standard error: 0.08959 on 4989 degrees of freedom
## Multiple R-squared:  0.8312, Adjusted R-squared:  0.8308
## F-statistic:  2456 on 10 and 4989 DF,  p-value: < 2.2e-16
```

From the summary, we can conclude that the linear model is

$$energy = 1.743 * e^{-1} - 2.253 * e^{-1} * acousticness - 1.078 * e^{-1} * danceability + \cdots + \epsilon$$

The adj R-squared value is 0.83 which is better than previous model. Let's check linearity of residual vs fitted line and normality of residuals.

Plots



Inference from the plots:

- **Residuals vs Fitted:** This plot is showing the residuals are having linear patterns.
- **Normal Q-Q plot:** This plot shows that residuals are not normally distributed.
- **Scale-Location plot:** This plot shows that Variance is almost constant.
- **Residual vs Leverage:** This plot talks about influential point.

Shapiro-Wilk normality test

```
##
##  Shapiro-Wilk normality test
##
## data:  Nlinear_model2$residuals
## W = 0.99038, p-value < 2.2e-16
```

From the output of the Shapiro test we can see that the p-value is less than 0.05. so, we say that the residuals are not normal. let's try another model.

Linear model

Model: energy ~ acousticness + loudness + valence

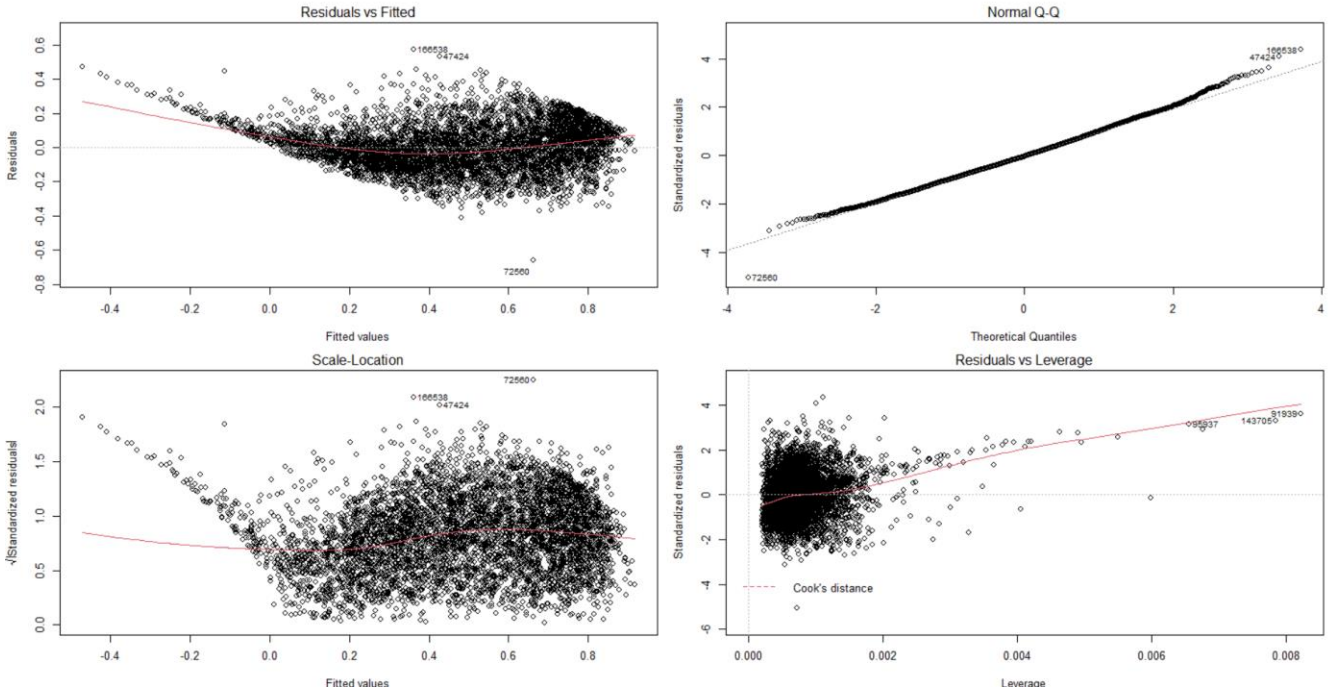
```
##
## Call:
## lm(formula = energy ~ acousticness + loudness + valence, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66195 -0.08802 -0.00591  0.08432  0.57216
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8743600  0.0065803  132.88   <2e-16 ***
## acousticness -0.3297852  0.0058564  -56.31   <2e-16 ***
## loudness      0.0238616  0.0003966   60.16   <2e-16 ***
## valence       0.1022318  0.0073262   13.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.131 on 4996 degrees of freedom
## Multiple R-squared:  0.7707, Adjusted R-squared:  0.7705
## F-statistic:  5596 on 3 and 4996 DF,  p-value: < 2.2e-16
```

From the summary, we can conclude that the linear model is

$$energy = 0.87436 - 0.32979 * acousticness + 0.02386 * loudness + 0.10223 * valence + \epsilon$$

The adj R-squared value is 0.77 which is less compared to previous model but let's check the normality and linearity.

Plots



Inference from the plots:

- **Residuals vs Fitted:** This plot is showing the residuals are not having linear patterns.
- **Normal Q-Q plot:** This plot shows that residuals are not normally distributed.
- **Scale-Location plot:** This plot shows that Variance is almost constant.
- **Residual vs Leverage:** This plot talks about influential point.

Shapiro-Wilk normality test

```
##
##  Shapiro-Wilk normality test
##
## data:  linear_model$residuals
## W = 0.99726, p-value = 7.246e-08

## [1] 0.00243076
```

From the output of the Shapiro test we can see that the p-value is less than 0.05. so, we say that the residuals are not normal. let's apply square root transformation on this model to get it normalized.

Normalized linear model

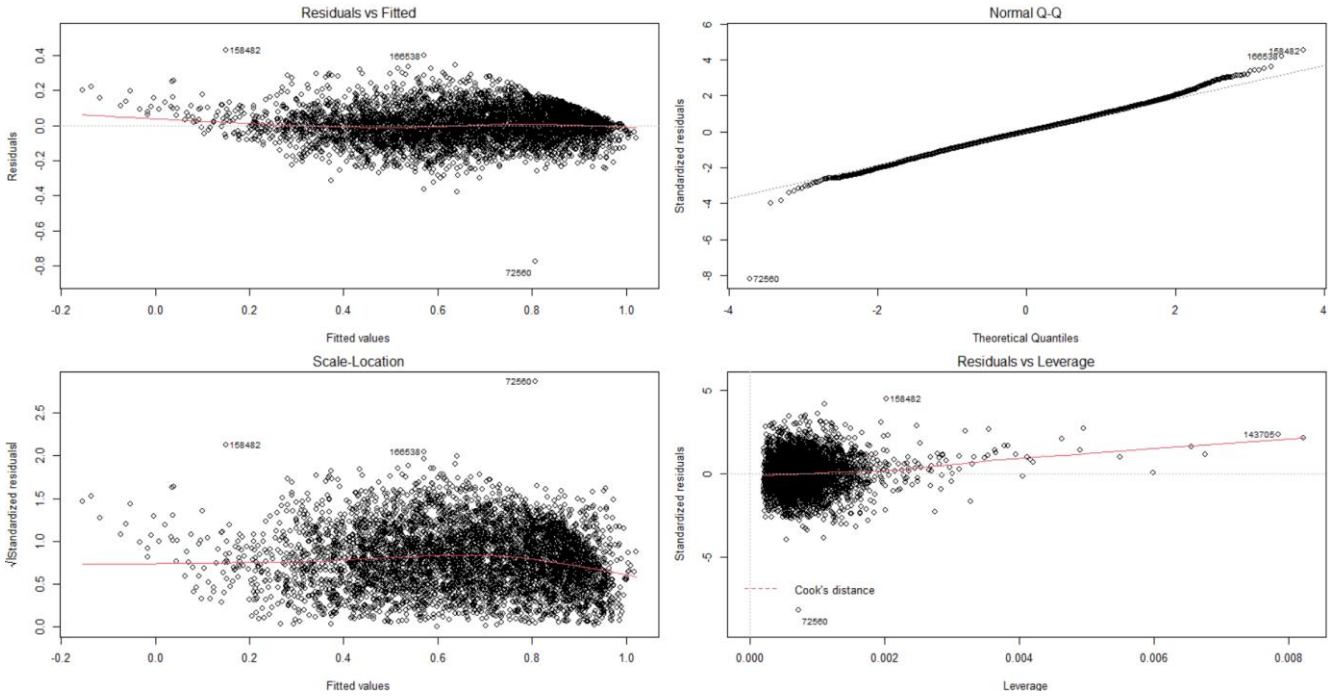
```
##
## Call:
## lm(formula = energyn ~ acousticness + loudness + valence, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77536 -0.05993  0.00047  0.05877  0.42665
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9661511  0.0047501  203.40   <2e-16 ***
## acousticness -0.2346286  0.0042276  -55.50   <2e-16 ***
## loudness      0.0210463  0.0002863   73.51   <2e-16 ***
## valence       0.1109111  0.0052886   20.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09459 on 4996 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8114
## F-statistic:  7169 on 3 and 4996 DF,  p-value: < 2.2e-16
```

From the summary, we can conclude that the linear model is

$$energy = 0.96615 - 0.23463 * acousticness + 0.02105 * loudness + 0.11091 * valence + \epsilon$$

The adj R-squared value is 0.81 which is much better compared to previous model. Now let's check the normality and linearity.

Plots



Inference from the plots:

- **Residuals vs Fitted:** This plot is showing the residuals are having linear patterns.
- **Normal Q-Q plot:** This plot shows that residuals are normally distributed with some outliers.
- **Scale-Location plot:** This plot shows that Variance is almost constant.
- **Residual vs Leverage:** This plot talks about influential point.

Shapiro-Wilk normality test

```
##
##  Shapiro-Wilk normality test
##
## data:  Nlinear_model$residuals
## W = 0.99384, p-value = 7.627e-14

## [1] 0.01571529
```

From the output of the Shapiro test we can see that the p-value is less than 0.05 but it is greater 0.001. so, we say that the residuals are normal at 1% significance level. Out of all models this model is giving normal residuals and better adj R-squared value. Now let's compare all the above models.

Comparison of Linear models

```
## # A tibble: 5 x 5
##   rowname         sigma adj.r.squared    AIC    BIC
##   <chr>           <dbl>         <dbl>  <dbl>  <dbl>
## 1 linear_model1  0.122          0.802 -6847. -6678.
```

```
## 2 linear_model2  0.122          0.801 -6832. -6754.
## 3 Nlinear_model2 0.0896         0.831 -9923. -9845.
## 4 linear_model   0.131          0.771 -6127. -6095.
## 5 Nlinear_model  0.0946         0.811 -9386. -9354.
```

From the above result we can see the sigma, adj R-squared, AIC and BIC values. Generally, the linear model is good if the AIC and BIC values the less and adj R-squared is above 0.70. In our case the adj R-squared is above 0.70 for all models but the AIC and BIC value are varying. AIC and BIC values for Normalized linear model (`Nlinear_model`) is very less compared to any other model excluding Nlinear_model2 because the residuals are not normal.

So, our model is energy ~ acousticness+loudness+valence. This is the only model which has normal residuals at 1% significance level. Therefore, the best model is given as:

$$energy = 0.96615 - 0.23463 * acousticness + 0.02105 * loudness + 0.11091 * valence + \epsilon$$

Conclusion: To reject or accept null hypothesis that, we defined in the beginning there should be statistical evidence and it is observed that the p-value of the Nlinear_model is less than $2.2{*}e^{-16}$ which is not statically significant with 5% or 1% significant level. so, we can reject the null hypothesis which is stating that all β_i are equal to zero.

Predictions with Nlinear_model on train data

```
##        actuals       fit       lwr       upr
## 155738   0.272 0.3143390 0.3089935 0.3197303
## 145934   0.447 0.3527687 0.3437552 0.3618988
## 168467   0.759 0.6526884 0.6426200 0.6628349
## 36293    0.382 0.3370821 0.3303452 0.3438870
## 52565    0.530 0.4731747 0.4682943 0.4780803
## 70790    0.528 0.5420540 0.5307091 0.5535189
```

The output shows the predicted values and actual values for the train data. It also shows the intervals of predicted data. From the predicted values it is observed that model is very good but it is not sufficient to infer that from train data. So, let's try the model on test data.

Predictions with Nlinear_model on test data

```
##        actuals        fit       lwr        upr
## 133294  0.3000 0.36670325 0.3605405 0.37291821
## 165838  0.8790 0.39450887 0.3852881 0.40383871
## 144026  0.2210 0.16302586 0.1582862 0.16783546
## 46844   0.4980 0.66371350 0.6566228 0.67084226
## 15911   0.7440 0.83309757 0.8228428 0.84341585
## 113219  0.2220 0.15987048 0.1540257 0.16582411
## 5462    0.4930 0.36410042 0.3570795 0.37118968
## 103952  0.6480 0.65036635 0.6390770 0.66175452
## 168365  0.0222 0.02252797 0.0193907 0.02590038
## 107518  0.9480 0.63289333 0.6189271 0.64701541
## 38152   0.9600 0.79689859 0.7857211 0.80815502
## 42429   0.1100 0.25500432 0.2489509 0.26113046
## 27019   0.1930 0.21202640 0.2071926 0.21691589
```

```
## 79901   0.5600 0.54652498 0.5317800 0.56147160
## 146905  0.3250 0.42208322 0.4167184 0.42748232
```

Previous table shows the predictions on known data but to check the model accuracy, let's predict on test data. This output shows the predicted values and actual values for the test data. It also shows the intervals of predicted data. From all these we can confirm that the data is fitting into the model which means the model is good.

Logistic regression

logistic model1

```
##
## Call:
## glm(formula = explicit ~ ., family = binomial(link = "logit"),
##     data = train[, -c(2, 11)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8654  -0.2204  -0.0736  -0.0194   4.7706
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -73.687361  10.989505  -6.705 2.01e-11 ***
## acousticness      -3.384874   0.400988  -8.441  < 2e-16 ***
## danceability       2.765505   0.567385   4.874 1.09e-06 ***
## energy            -2.558085   0.610228  -4.192 2.76e-05 ***
## instrumentalness  -0.401796   0.398872  -1.007   0.3138
## key1               0.199360   0.289951   0.688   0.4917
## key2               0.186150   0.316051   0.589   0.5559
## key3              -0.592301   0.542772  -1.091   0.2752
## key4               0.153309   0.336885   0.455   0.6491
## key5              -0.346295   0.363287  -0.953   0.3405
## key6              -0.122763   0.323941  -0.379   0.7047
## key7              -0.436345   0.326844  -1.335   0.1819
## key8               0.042598   0.359438   0.119   0.9057
## key9              -0.528083   0.362488  -1.457   0.1452
## key10             -0.289403   0.368856  -0.785   0.4327
## key11              0.021616   0.315927   0.068   0.9455
## liveness           0.614918   0.416877   1.475   0.1402
## loudness           0.230895   0.034642   6.665 2.64e-11 ***
## mode1             -0.160852   0.158258  -1.016   0.3094
## popularity         0.022439   0.003706   6.055 1.41e-09 ***
## speechiness        8.792070   0.529498  16.605  < 2e-16 ***
## tempo             -0.001581   0.002553  -0.619   0.5357
## valence           -1.974633   0.372782  -5.297 1.18e-07 ***
## year               0.036782   0.005439   6.763 1.35e-11 ***
## duration_min      -0.113306   0.054201  -2.090   0.0366 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2410.4  on 4999  degrees of freedom
```

```
## Residual deviance: 1322.5  on 4975  degrees of freedom
## AIC: 1372.5
##
## Number of Fisher Scoring iterations: 8
```

we can infer the following from the above output:

- This model contains many non-significant variables which will add penalty to this model.
- The AIC value of this model is 1372.5.

Let's create another model with only significant variables and try to achieve a lower AIC value.

logistic model

```
##
## Call:
## glm(formula = explicit ~ acousticness + danceability + energy +
##     loudness + popularity + speechiness + valence + year + duration_min,
##     family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6380  -0.2251  -0.0742  -0.0196   4.7863
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -73.527754  10.789215  -6.815 9.43e-12 ***
## acousticness  -3.325968   0.392411  -8.476  < 2e-16 ***
## danceability   2.862712   0.544548   5.257 1.46e-07 ***
## energy        -2.440306   0.589012  -4.143 3.43e-05 ***
## loudness       0.239239   0.033733   7.092 1.32e-12 ***
## popularity     0.023326   0.003447   6.766 1.32e-11 ***
## speechiness    9.101699   0.501018  18.166  < 2e-16 ***
## valence       -2.002095   0.361267  -5.542 2.99e-08 ***
## year           0.036485   0.005343   6.829 8.55e-12 ***
## duration_min  -0.113494   0.053448  -2.123   0.0337 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2410.4  on 4999  degrees of freedom
## Residual deviance: 1338.7  on 4990  degrees of freedom
## AIC: 1358.7
##
## Number of Fisher Scoring iterations: 8
```

we achieved a lower AIC value and a better model. And also, we are not having any non-significant variables. Now, we can compare both the models using the ANOVA test.

Let's say our null hypothesis and alternative hypothesis with 5% significance level are as follows:

H_0: second model is better than the first model.

H_1: first model is better than the second model

Comparing two models using anova

```
## Analysis of Deviance Table
##
## Model 1: explicit ~ acousticness + danceability + energy + instrumentalness +
##     key + liveness + loudness + mode + popularity + speechiness +
##     tempo + valence + year + duration_min
## Model 2: explicit ~ acousticness + danceability + energy + loudness +
##     popularity + speechiness + valence + year + duration_min
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      4975     1322.5
## 2      4990     1338.7 -15  -16.146   0.3724
```

We can see from the above results that the p-value is greater than 0.05, indicating that the second model is better to the first model. Let's take a look at how well the logistic model performed on both train and test results. Since the second model is superior, we will use it to make predictions.

Predictions on train data

```
## Confusion Matrix and Statistics
##
##          actual
## Predicted    0    1
##         0 4109   55
##         1  565  271
##
##                Accuracy : 0.876
##                  95% CI : (0.8665, 0.885)
##     No Information Rate : 0.9348
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.4112
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8313
##             Specificity : 0.8791
##          Pos Pred Value : 0.3242
##          Neg Pred Value : 0.9868
##              Prevalence : 0.0652
##          Detection Rate : 0.0542
##    Detection Prevalence : 0.1672
##       Balanced Accuracy : 0.8552
##
##        'Positive' Class : 1
##
```

We are able to accurately predict 4109 out of 4674 "No" cases and 271 out of 326 "Yes" cases. This suggests that logistic regression will predict "No" cases about 83% and "Yes" cases about 87.9%, with an overall accuracy of 87.6% on train data.

Predictions on test data

```
## Confusion Matrix and Statistics
##
##          actual
## predicted    0    1
##         0 1664   27
##         1  208  101
##
##                Accuracy : 0.8825
##                  95% CI : (0.8676, 0.8963)
##     No Information Rate : 0.936
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.4087
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7891
##             Specificity : 0.8889
##          Pos Pred Value : 0.3269
##          Neg Pred Value : 0.9840
##              Prevalence : 0.0640
##          Detection Rate : 0.0505
##    Detection Prevalence : 0.1545
##       Balanced Accuracy : 0.8390
##
##        'Positive' Class : 1
##
```

We are able to accurately distinguish 1664 of 1872 "No" cases and 101 of 128 "Yes" cases. This s uggests that logistic regression will predict "No" cases about 78.9 % and "Yes" cases about 88.8 %, with an overall accuracy of 88 % on test data.

Naive Bayes classifier

Naive Bayes model

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##      0      1
## 0.9348 0.0652
##
## Conditional probabilities:
```

```
##    acousticness
## Y        [,1]      [,2]
##   0 0.5114419 0.3778808
##   1 0.1902600 0.2253049
##
##    danceability
## Y        [,1]      [,2]
##   0 0.5249582 0.1735104
##   1 0.6593098 0.1674543
##
##    energy
## Y        [,1]      [,2]
##   0 0.4797514 0.2729943
##   1 0.5944767 0.2589146
##
##    loudness
## Y         [,1]     [,2]
##   0 -11.864579 5.753485
##   1  -9.246525 5.618809
##
##    popularity
## Y       [,1]     [,2]
##   0 25.55691 21.26606
##   1 37.28528 27.90304
##
##    speechiness
## Y         [,1]      [,2]
##   0 0.08835689 0.1545888
##   1 0.34047209 0.3198956
##
##    valence
## Y        [,1]      [,2]
##   0 0.5263477 0.2691587
##   1 0.5047681 0.2133967
##
##    year
## Y       [,1]     [,2]
##   0 1976.408 26.01666
##   1 1992.684 30.01704
##
##    duration_min
## Y       [,1]     [,2]
##   0 3.915194 2.575604
##   1 3.440002 1.294014
```

The Naive Bayes model generates separate prior probabilities for the response class and conditional probabilities for each attribute. The prior probability of 0 is 93%, and the prior probability of 1 is 6%. For the response variable, we can see the conditional probability of each attribute. Let us now examine the model's performance on both train and test data.

Predictions on train data

```
## Confusion Matrix and Statistics
##
##          actual
```

```
## predicted    0    1
##         0 4363  126
##         1  311  200
##
##                Accuracy : 0.9126
##                  95% CI : (0.9044, 0.9203)
##     No Information Rate : 0.9348
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.4327
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9335
##             Specificity : 0.6135
##          Pos Pred Value : 0.9719
##          Neg Pred Value : 0.3914
##              Prevalence : 0.9348
##          Detection Rate : 0.8726
##    Detection Prevalence : 0.8978
##       Balanced Accuracy : 0.7735
##
##        'Positive' Class : 0
##
```

We are able to accurately identify 4363 out of 4674 "No" cases and 200 out of 326 "Yes" cases. This suggests that the Naive Bayes algorithm can predict “No” cases with an accuracy of about 93 percent, but only 61 percent of “Yes” cases, resulting in an overall accuracy of 91 percent on train data. Now let's check the model on test data.

Predictions on test data

```
## Confusion Matrix and Statistics
##
##          actual
## predicted    0    1
##         0 1749   44
##         1  123   84
##
##                Accuracy : 0.9165
##                  95% CI : (0.9035, 0.9283)
##     No Information Rate : 0.936
##     P-Value [Acc > NIR] : 0.9997
##
##                   Kappa : 0.4587
##
##  Mcnemar's Test P-Value : 1.582e-09
##
##             Sensitivity : 0.9343
##             Specificity : 0.6562
##          Pos Pred Value : 0.9755
##          Neg Pred Value : 0.4058
##              Prevalence : 0.9360
##          Detection Rate : 0.8745
##    Detection Prevalence : 0.8965
```

```
##       Balanced Accuracy : 0.7953
##
##        'Positive' Class : 0
##
```

We are able to accurately distinguish 1749 out of 1872 "No" cases and 84 out of 128 "Yes" cases. This suggests that the Naive Bayes algorithm can predict “No” cases with an accuracy of about 93 percent, but it can only predict “Yes” cases with an accuracy of 65.6 percent, resulting in an overall accuracy of 91.6 percent on test data.

Linear Discrimination Analysis

Let’s classify explicit by linear discrimination analysis. First check for the constant covariance assumption.

Test for covariance assumption

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  train[, -c(2, 5, 7, 10, 11, 18)]
## Chi-Sq (approx.) = 1848.7, df = 78, p-value < 2.2e-16
```

The above result says that the assumption is not full-filled because the p-value is not significant. Since, the assumption is failed let’s do quadratic discrimination analysis on explicit.

Quadratic Discrimination Analysis

QDA model

```
## Call:
## qda(explicit ~ acousticness + danceability + energy + loudness +
##     popularity + speechiness + valence + year + duration_min,
##     data = train)
##
## Prior probabilities of groups:
##      0      1
## 0.9348 0.0652
##
## Group means:
##   acousticness danceability    energy   loudness popularity speechiness
## 0    0.5114419    0.5249582 0.4797514 -11.864579   25.55691  0.08835689
## 1    0.1902600    0.6593098 0.5944767  -9.246525   37.28528  0.34047209
##     valence     year duration_min
## 0 0.5263477 1976.408     3.915194
## 1 0.5047681 1992.684     3.440002
```

The above results show prior probabilities same as Naive Bayes. It is also showing the group wise means of each individual variable. Now let’s observe the model accuracy on train and test data.

Predictions on train data

```
## Confusion Matrix and Statistics
##
##          actual
## predicted    0    1
##         0 4443  141
##         1  231  185
##
##                Accuracy : 0.9256
##                  95% CI : (0.918, 0.9327)
##     No Information Rate : 0.9348
##     P-Value [Acc > NIR] : 0.9956
##
##                   Kappa : 0.4591
##
##  Mcnemar's Test P-Value : 3.942e-06
##
##             Sensitivity : 0.9506
##             Specificity : 0.5675
##          Pos Pred Value : 0.9692
##          Neg Pred Value : 0.4447
##              Prevalence : 0.9348
##          Detection Rate : 0.8886
##    Detection Prevalence : 0.9168
##       Balanced Accuracy : 0.7590
##
##        'Positive' Class : 0
##
```

We are able to accurately distinguish 4433 out of 4674 "No" cases and 185 out of 326 "Yes" cases. This suggests that quadratic discrimination analysis ability to predict "No" cases are approximately 95%, but it falls to 57% for "Yes" cases, resulting in an overall accuracy of 92.5% on train data. Now let's check the model on test data.

Predictions on test data

```
## Confusion Matrix and Statistics
##
##          actual
## predicted    0    1
##         0 1781   59
##         1   91   69
##
##                Accuracy : 0.925
##                  95% CI : (0.9126, 0.9362)
##     No Information Rate : 0.936
##     P-Value [Acc > NIR] : 0.97811
##
##                   Kappa : 0.4393
##
##  Mcnemar's Test P-Value : 0.01137
##
##             Sensitivity : 0.9514
##             Specificity : 0.5391
```

```
##          Pos Pred Value : 0.9679
##          Neg Pred Value : 0.4312
##              Prevalence : 0.9360
##          Detection Rate : 0.8905
##    Detection Prevalence : 0.9200
##       Balanced Accuracy : 0.7452
##
##        'Positive' Class : 0
##
```

We are able to accurately distinguish 1781 out of 1872 "No" cases and 69 out of 128 "Yes" cases. This suggests that quadratic discrimination analysis potential to predict "No" cases are approximately 95%, but it falls to 54% of "Yes" cases, resulting in an overall accuracy of 92.5 percent on test data.

Multinomial Logistic regression

Multinominal model

```
## Loading required package: nnet

## # weights:  11 (10 variable)
## initial  value 3465.735903
## iter  10 value 772.490500
## iter  20 value 697.118637
## iter  30 value 679.773898
## final  value 669.335102
## converged

## Call:
## multinom(formula = explicit ~ acousticness + danceability + energy +
##     loudness + popularity + speechiness + valence + year + duration_min,
##     data = train)
##
## Coefficients:
##                     Values    Std. Err.
## (Intercept)  -73.52876170 0.0020333466
## acousticness  -3.32603921 0.0667312292
## danceability   2.86272362 0.1129646134
## energy        -2.44030350 0.0316690389
## loudness       0.23923788 0.0168093606
## popularity     0.02332624 0.0032449554
## speechiness    9.10176339 0.0684117941
## valence       -2.00210688 0.2478672548
## year           0.03648524 0.0001799801
## duration_min  -0.11349048 0.0520067189
##
## Residual Deviance: 1338.67
## AIC: 1358.67
```

We can observe that, even after assigning the model to a new entity, we can see that some output is created by running the model. This model's running output contains some iteration history as well as the final negative log-likelihood 669.335. This value is multiplied by two and appears as

the Residual Deviance in the model description, which is 1338.67. The summary of the model has a block of coefficients and a block of standard errors.

Predictions on train data

```
## Confusion Matrix and Statistics
##
##          actual
## predicted    0    1
##         0 4629  197
##         1   45  129
##
##                Accuracy : 0.9516
##                  95% CI : (0.9453, 0.9574)
##     No Information Rate : 0.9348
##     P-Value [Acc > NIR] : 3.041e-07
##
##                   Kappa : 0.493
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9904
##             Specificity : 0.3957
##          Pos Pred Value : 0.9592
##          Neg Pred Value : 0.7414
##              Prevalence : 0.9348
##          Detection Rate : 0.9258
##    Detection Prevalence : 0.9652
##       Balanced Accuracy : 0.6930
##
##        'Positive' Class : 0
##
```

We are able to accurately recognize 4629 out of 4674 "No" cases and 129 out of 326 "Yes" cases. This suggests that the Multinominal model's ability to predict "No" cases is nearly 99 percent, but it falls drastically to 39 percent of "Yes" cases, resulting in an overall accuracy of 95 percent on train data. Now let's check the model on test data.

Predictions on test data

```
## Confusion Matrix and Statistics
##
##          actual
## predicted    0    1
##         0 1850   77
##         1   22   51
##
##                Accuracy : 0.9505
##                  95% CI : (0.9401, 0.9596)
##     No Information Rate : 0.936
##     P-Value [Acc > NIR] : 0.003577
##
##                   Kappa : 0.4834
##
##  Mcnemar's Test P-Value : 5.724e-08
```

```
##
##             Sensitivity : 0.9882
##             Specificity : 0.3984
##          Pos Pred Value : 0.9600
##          Neg Pred Value : 0.6986
##              Prevalence : 0.9360
##          Detection Rate : 0.9250
##    Detection Prevalence : 0.9635
##       Balanced Accuracy : 0.6933
##
##        'Positive' Class : 0
##
```

We are able to accurately recognize 1850 of 1872 "No" cases and 77 of 128 "Yes" cases. This suggests that the Multinominal model's ability to predict "No" cases is nearly 99 percent, but it falls drastically to 40 percent of "Yes" cases, resulting in an overall accuracy of 95 percent on test data.

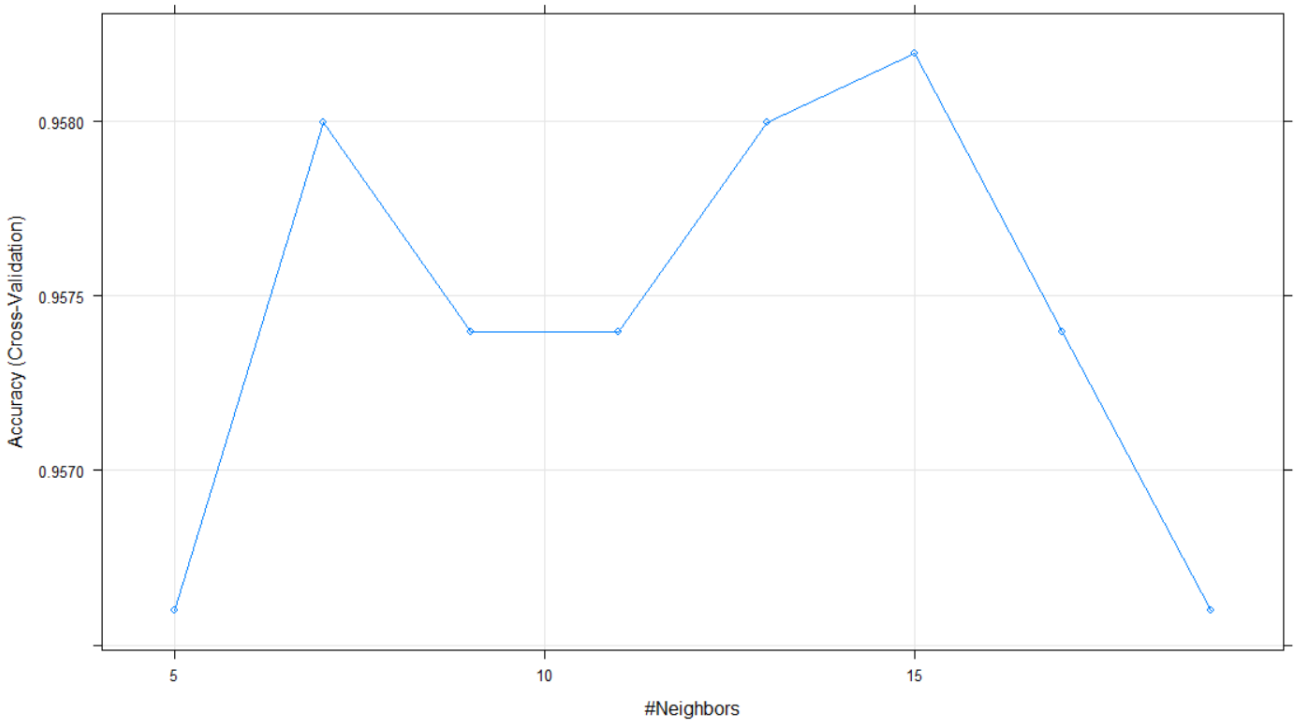
k-Nearest Neighbors

k-NN model

```
## k-Nearest Neighbors
##
## 5000 samples
##    9 predictor
##    2 classes: '0', '1'
##
## Pre-processing: centered (9), scaled (9)
## Resampling: Cross-Validated (8 fold)
## Summary of sample sizes: 4375, 4375, 4374, 4374, 4375, 4376, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.9565977  0.5523164
##    7  0.9579987  0.5632166
##    9  0.9573967  0.5586668
##   11  0.9573980  0.5577665
##   13  0.9579977  0.5599037
##   15  0.9581971  0.5556234
##   17  0.9573990  0.5469425
##   19  0.9565990  0.5355998
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 15.
```

KNN, or K Nearest Neighbor, is a Supervised Machine Learning algorithm that classifies a new data point into the target class based on its neighboring data points characteristics. The output above shows the model's iterations, and it choose 15 as the best k value for the model.

plot



This plot showing the k=15 is best for getting good accuracy.

Predictions on train data

```
## Confusion Matrix and Statistics
##
##          actual
## predicted    0    1
##         0 4655  177
##         1   19  149
##
##                Accuracy : 0.9608
##                  95% CI : (0.955, 0.966)
##     No Information Rate : 0.9348
##     P-Value [Acc > NIR] : 8.182e-16
##
##                   Kappa : 0.5848
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9959
##             Specificity : 0.4571
##          Pos Pred Value : 0.9634
##          Neg Pred Value : 0.8869
##              Prevalence : 0.9348
##          Detection Rate : 0.9310
##    Detection Prevalence : 0.9664
##       Balanced Accuracy : 0.7265
##
```

```
##        'Positive' Class : 0
##
```

We are able to accurately distinguish 4655 of 4674 "No" cases and 149 of 326 "Yes" cases. This suggests that KNN's ability to forecast "No" cases are approximately 99.5 percent, but it falls to 45.7 percent of "Yes" cases, resulting in an overall accuracy of 96 percent on train data. Now let's check the model on test data.

Predictions on test data

```
## Confusion Matrix and Statistics
##
##          actual
## predicted    0    1
##         0 1852   71
##         1   20   57
##
##                Accuracy : 0.9545
##                  95% CI : (0.9444, 0.9632)
##     No Information Rate : 0.936
##     P-Value [Acc > NIR] : 0.0002419
##
##                   Kappa : 0.5337
##
##  Mcnemar's Test P-Value : 1.593e-07
##
##             Sensitivity : 0.9893
##             Specificity : 0.4453
##          Pos Pred Value : 0.9631
##          Neg Pred Value : 0.7403
##              Prevalence : 0.9360
##          Detection Rate : 0.9260
##    Detection Prevalence : 0.9615
##       Balanced Accuracy : 0.7173
##
##        'Positive' Class : 0
##
```

We are able to accurately distinguish 1852 of 1872 "No" cases and 57 of 128 "Yes" cases. This suggests that KNN's potential to forecast "No" cases are approximately 99 percent, but it falls drastically to 44.5 percent of "Yes" cases, resulting in an overall accuracy of 95 percent on test data.

Random forest

Random forest model

```
##
## Call:
##  randomForest(formula = explicit ~ acousticness + danceability +      energy +
loudness + popularity + speechiness + valence +      year + duration_min, data = t
rain, ntree = 600, mtry = 4,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 600
```

```
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 4.26%
## Confusion matrix:
##      0   1 class.error
## 0 4638  36 0.007702182
## 1  177 149 0.542944785
```

The above output shows that type of random forest is classification and number of trees used are 600. The error rate is 4.26% and class wise error is also given. Now check the model predictions on train and test data.

Predictions on train data

```
## Confusion Matrix and Statistics
##
##
## predTrain    0    1
##         0 4674    0
##         1    0  326
##
##                Accuracy : 1
##                  95% CI : (0.9993, 1)
##     No Information Rate : 0.9348
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.9348
##          Detection Rate : 0.9348
##    Detection Prevalence : 0.9348
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

We are able to accurately distinguish 4674 of 4674 "No" cases and 326 of 326 "Yes" cases. This suggests that Random forest ability to forecast "No" cases are 100 and "Yes" cases are also 100%, resulting in an overall accuracy of 100 percent on train data. Now let's check the model on test data.

Predictions on test data

```
## Confusion Matrix and Statistics
##
##
## predTest    0    1
##        0 1852   63
##        1   20   65
```

```
##
##                Accuracy : 0.9585
##                  95% CI : (0.9488, 0.9668)
##     No Information Rate : 0.936
##     P-Value [Acc > NIR] : 8.121e-06
##
##                   Kappa : 0.5894
##
##  Mcnemar's Test P-Value : 4.025e-06
##
##             Sensitivity : 0.9893
##             Specificity : 0.5078
##          Pos Pred Value : 0.9671
##          Neg Pred Value : 0.7647
##              Prevalence : 0.9360
##          Detection Rate : 0.9260
##    Detection Prevalence : 0.9575
##       Balanced Accuracy : 0.7486
##
##        'Positive' Class : 0
##
```

We are able to accurately distinguish 1852 of 1872 "No" cases and 63 of 128 "Yes" cases. This suggests that Random forest potential to forecast "No" cases are approximately 99 percent, but it falls drastically to 50 percent of "Yes" cases, resulting in an overall accuracy of 95.8 percent on test data.

Comparison of classification models

Train data

```
## [1] "Confusion Matrix for train data "

## [1] "Logistic Quadratic Naive_Bayes"

##      0   1    0   1    0   1
## 0 4109  55 4443 141 4363 126
## 1  565 271  231 185  311 200

## [1] "K-NN Multinominal Random_forest"

##      0   1    0   1    0   1
## 0 4655 177 4629 197 4674   0
## 1   19 149   45 129    0 326
```

The above output shows the confusion Matrix of various models when the model is applied on train data. We can observe the Random forest is giving 100% accuracy which means it is overfitting. Now, logistic regression is predicating sensitivity as 83% and specificity as 88% whereas all other models are biased towards “No” classes. But applying model on train data is of no use sometimes (see Random forest which is overfitting) because the data is seen while training. So, let’s see models that applied on test data.

Test data

```
## [1] "Confusion Matrix for test data "

## [1] "Logistic Quadratic Naive Bayes"

##      0   1    0  1    0  1
## 0 1664  27 1781 59 1749 44
## 1  208 101   91 69  123 84

## [1] "K-NN Multinominal Random_forest"

##      0  1    0  1    0  1
## 0 1852 71 1850 77 1852 63
## 1   20 57   22 51   20 65
```

The above output shows the confusion Matrix of different models when the model is applied on test data. Here Random forest is giving 95% accuracy and also its specificity is reduced almost by 50% whereas it is giving 100% accuracy in train data. Now, logistic regression is predicating sensitivity as 79% and specificity as 88% which is still maintaining the balance whereas all other models are biased towards "No" classes same as train data. From all the inferences above we can make out logistic regression model as best classifier.

Conclusion

We assimilated an abundance of knowledge from all the plots in exploratory data analysis. First, we started with an unsupervised method namely, principal component analysis which is a technique for extracting essential variables in the form of components from a wide number of variables in a data set. In our case, eight principal components are required to account for 90% of the variance in the data which is determined by the scree plot.

After that, we begin to separate the data using an unsupervised algorithm called k-means. The K-Means algorithm's main goal is to minimize the sum of distances between points and their respective cluster centroid. Using this algorithm, we obtained various clusters with different k values; from these, we chose k as 4, which is defined by the elbow curve.

Further we went on to supervised algorithms such as linear regression, logistic regression, and so on. Using exploratory data analysis, we applied linear regression with energy as the response variable, yielding a good adjusted R-Squared with only three significant variables among a few models. so, our best model is energy ~ acousticness+loudness+valence.

Following that, we commenced classifying explicit and non-explicit musical compositions from the data. For this, we endeavored different models like logistic regression, Naive Bayes, Multinominal model, Quadratic discrimination analysis, K-Nearest Neighbors, and Random forest. Among, all the models we found that logistic regression is giving better accuracy with the best sensitivity and specificity values.

References

https://cran.r-project.org/

https://rpubs.com/

https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks

https://www.tutorialspoint.com/r/index.htm

https://docs.rstudio.com/

https://www.wikipedia.org/

https://ggplot2.tidyverse.org/reference/ggplot.html